

NASA Technical Memorandum 4440

IN 08
175538
253

A High-Fidelity,
Six-Degree-of-Freedom
Batch Simulation
Environment for
Tactical Guidance
Research and Evaluation

Kenneth H. Goodrich

JULY 1993

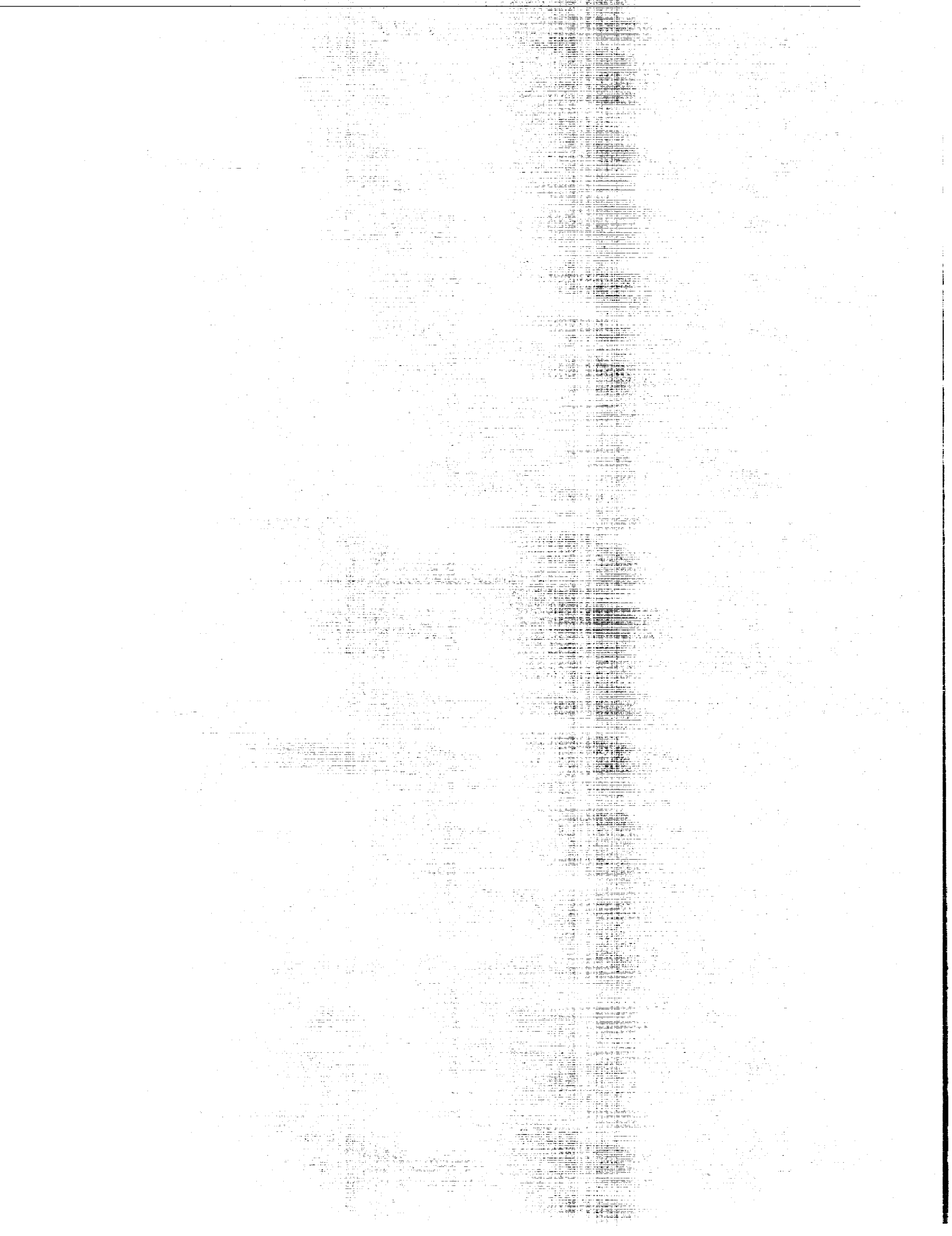
(NASA-TM-4440) A HIGH-FIDELITY,
SIX-DEGREE-OF-FREEDOM BATCH
SIMULATION ENVIRONMENT FOR TACTICAL
GUIDANCE RESEARCH AND EVALUATION
(NASA) 53 p

N93-32380

Unclass

H1/08 0175538

NASA



A High-Fidelity,
Six-Degree-of-Freedom
Batch Simulation
Environment for
Tactical Guidance
Research and Evaluation

Kenneth H. Goodrich
Langley Research Center
Hampton, Virginia



National Aeronautics and
Space Administration

Office of Management

Scientific and Technical
Information Program

1993

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Contents

Abstract	1
Introduction	1
Symbols and Abbreviations	2
Background and Objectives	3
The Tactical Maneuvering Simulator	5
Functional Overview	5
Aircraft Simulation Model	6
Implementation of simulation	6
Equations of motion	6
Aerodynamic forces and moments	7
Engine forces and moments	7
Control augmentation system	8
Tactical Autopilot	9
Multiple Aircraft Simulation and TMS Executive Program	12
Demonstration of the Tactical Maneuvering Simulator	13
Simulation of Four Aircraft	13
One-Versus-One Engagement	14
Future Research Activities	15
Concluding Remarks	15
References	16
Appendix A—TMS Executive Program	18
Appendix B—Communication and Synchronization Subroutine	21
Appendix C—Example Trim File	24
Tables	27
Figures	31

Abstract

A batch air combat simulation environment, the tactical maneuvering simulator (TMS), is a tool for developing and evaluating tactical maneuvering logics that can also be used to evaluate the tactical implications of perturbations to aircraft performance or supporting systems. The TMS can simulate air combat between any number of engagement participants, with practical limits imposed by computer memory and processing power. Aircraft are modeled using equations of motion, control laws, aerodynamics, and propulsive characteristics equivalent to those used in high-fidelity piloted simulation. Data bases representative of a modern high-performance aircraft with and without thrust-vectoring capability are included. To simplify the task of developing and implementing maneuvering logics in the TMS, an outer-loop control system, the tactical autopilot (TA), is implemented in the aircraft simulation model. The TA converts guidance commands by computerized maneuvering logics from desired angle of attack and wind-axis bank angle to inputs for the inner-loop control augmentation system of the aircraft. This report describes the capabilities and operation of the TMS and the TA.

Introduction

As new technologies or capabilities are proposed for high-performance aircraft, the impact, utilization, and costs of these technologies must be assessed within the context of air combat tactics and effectiveness. The highly complex and transient nature of air combat makes simulation the primary tool for performing this assessment. Both batch and real-time, piloted simulations can contribute to the assessment.

Batch air combat simulations such as the advanced air-to-air system performance model (ref. 1) and TAC BRAWLER (ref. 2) allow the study of aircraft tactics and performance in a highly controlled and repeatable environment. Batch air combat simulations consist of two fundamental elements—computerized maneuvering logics that generate maneuver decisions and a simulation environment in which maneuvering logics are developed and tested. Batch combat simulation programs can run large numbers of engagements with minimal operator intervention, which allows comprehensive sets of initial conditions or parametric variations to be rapidly evaluated. Unfortunately, the minimal operator intervention inherent in batch operation slows development and validation of new maneuvering logics, which can result in relatively inflexible tactics that do not effectively exploit a given situation or aircraft capability.

In contrast, piloted simulation provides an environment ideally suited for rapid tactical experimentation and adaptation. New tactics can be investigated

by instructing pilots to maneuver in the desired manner. Furthermore, the natural interface provided to the pilots encourages their participation in this development process and enhances their ability to assess the success of a given tactic. Unfortunately, because human pilots introduce variability, the time required to perform a statistically meaningful piloted air combat simulation study, combined with the availability and expense of the necessary facilities and pilots, makes a comprehensive study extremely difficult to perform.

Because the strengths and weaknesses of batch and piloted simulations are complementary, a synergism exists when the two approaches are employed in concert. To fully exploit this synergy, the Langley Research Center is developing an integrated batch and piloted simulation tool known as the tactical guidance research and evaluation system (known as TiGRES in 1989 when ref. 3 was written). The TiGRES tool consists of three primary elements: an advanced maneuvering logic that functions in real time and uses artificial intelligence techniques (ref. 4); a multidome, piloted simulation facility, the differential maneuvering simulator (DMS, ref. 5); and a batch simulation environment, the tactical maneuvering simulator (TMS). The development and operation of the TMS and its relation to the other elements of the TiGRES tool are the focuses of this report.

Unlike existing batch air combat simulation environments that typically use reduced-order dynamic models, aircraft in the TMS are modeled using

equations of motion, control laws, aerodynamics, and propulsive characteristics identical to those used in high-fidelity piloted simulations in the DMS. This commonality allows maneuvering logics developed in the TMS to be evaluated without modification in relation to human pilots in the DMS. The ability to test maneuvering logics with human pilots provides an efficient means of validating the results of batch simulation analysis. Thus, extensive preliminary investigations of tactical maneuvering strategies, guidance concepts, or aircraft performance characteristics can be performed quickly and cheaply with the TMS. After the focus of an investigation matures, a minimum number of piloted simulations in the DMS can confirm or refine the findings of the more comprehensive batch analysis.

The TMS has three basic elements. The first element is the model that simulates individual aircraft. Currently, models representative of a modern high-performance aircraft with and without thrust-vectoring (TV) capability are available. The second element is the tactical autopilot (TA), which enables maneuvering logics to command full-order dynamic aircraft models in both the TMS and DMS. The TA converts guidance commands issued in the form of desired angle of attack and wind-axis bank angle into inputs to the inner-loop control augmentation system of the simulated aircraft. The third element is the TMS executive program and the synchronization subroutine; these provide the capability to simulate many-versus-many (MvN) air combat by running multiple, single-aircraft simulations in parallel.

This report describes the capabilities and operation of the TMS. First, the background underlying the development of the TMS is discussed. Next, the simulation environment is described. This description details the available aircraft models, the TA, and the parallel implementation used to provide MvN simulation. Thereafter, example engagements are presented to demonstrate TMS operation. The paper concludes with a discussion of future areas of research and a summary of the current work.

Symbols and Abbreviations

Symbols:

C_L	lift coefficient, Lift/ $(\bar{q} \times \text{Reference area})$
\bar{c}	mean aerodynamic chord
F_X, F_Y, F_Z	force about X -, Y -, and Z -axes, lb
g	acceleration due to gravity, 32.17 ft/sec ²

h	altitude, ft
I_X	rolling moment of inertia, slug-ft ²
I_{XZ}	product of inertia, slug-ft ²
I_Y	pitching moment of inertia, slug-ft ²
I_Z	yawing moment of inertia, slug-ft ²
$K_{D\alpha}$	gain on rate of α error
$K_{D\mu}$	gain on rate of μ error
$K_{I\alpha}$	gain on integral of α error
$K_{P\alpha}$	proportional gain on α error
$K_{P\mu}$	proportional gain on μ error
\mathbf{L}_{BE}	transfer motion matrix from Earth to body axis
\mathbf{L}_{WB}	transfer motion matrix from body to wind axis
\mathbf{L}_{WE}	transfer motion matrix from Earth to wind axis
M	Mach number
M_X, M_Y, M_Z	moment about X -, Y -, and Z -axes, ft-lb
M_P	maximum peak overshoot
m	aircraft mass, slugs
n	normal load factor, g units
p	roll rate in body-axis system, deg/sec
q	pitch rate in body-axis system, deg/sec
\bar{q}	dynamic pressure, lb/ft ²
r	yaw rate in body-axis system, deg/sec
s	Laplace operator
TH	body-axis components of thrust force, lb
t	time, sec
th	thrust force, lb
u	velocity along X body axis, ft/sec
v	velocity along Y body axis, ft/sec
w	velocity along Z body axis, ft/sec

X, Y, Z	longitudinal, lateral, and vertical body axes
X_E	X -axis of inertial reference system
X_{eng}	separation along X -axis between center of gravity and thrust force line of action
Y_E	Y -axis of inertial reference system
Y_{eng}	separation along Y -axis between center of gravity and thrust force line of action
Z_E	Z -axis of inertial reference system
Z_{eng}	separation along Z -axis between center of gravity and thrust force line of action
α	angle of attack, deg
$\dot{\alpha}$	rate of change of α , deg/sec
β	angle of sideslip, deg
γ	flight path angle, deg
δ_{azimTV}	change in azimuth angle due to thrust vectoring, deg
δ_{elevTV}	change in elevation angle due to thrust vectoring, deg
δ_{lat}	lateral stick displacement, in.
δ_{lon}	longitudinal stick displacement, in.
δ_{th}	thrust deflection angle, deg
θ	body-axis pitch angle, deg
μ	wind-axis bank angle, deg
$\dot{\mu}$	rate of change of μ , deg/sec
ρ/ρ_0	density ratio
ρ_m	maneuver plane rotation angle, deg
ϕ	body-axis bank angle, deg
ψ	body-axis heading angle, deg
Subscripts:	
A	aerodynamic
a	aileron
E	engine

H	horizontal stabilator
L	left engine
R	right engine
r	rudder
REF	reference
SB	speedbrake
s	stabilator
LEF	leading-edge flap
TEF	trailing-edge flap
Abbreviations:	
ACM	air combat maneuvering
ACSL	Advanced Continuous Simulation Language
AML	Adaptive Maneuvering Logic
azim ₀	engine azimuth angle as mounted to airframe, deg
CAS	control augmentation system
DMS	Differential Maneuvering Simulator
d.o.f.	degrees of freedom
elev ₀	engine elevation angle as mounted to airframe, deg
MvN	many versus many
TA	tactical autopilot
TDG	tactical decision generator
TiGRES	tactical guidance research and evaluation system
TMS	tactical maneuvering simulator
TV	thrust vectored
1v1	one versus one

Background and Objectives

During the late 1960's and 1970's, NASA funded the development of a computer program to provide an invariant or calibrated opponent for use in piloted air combat simulation studies in the newly constructed DMS. (See ref. 6.) The original specification called for a program capable of generating tactically sound maneuver decisions and of realistically simulating the resulting aircraft motions for an arbitrary aircraft in one-versus-one (1v1) air combat. Researchers recognized that such a program would not only provide an invariant opponent in the DMS, but could also be used to perform rapid parametric studies on different aircraft characteristics and to develop

new tactical maneuvers for existing and proposed aircraft. A final requirement was for the program to run in real time on the computer system of the DMS (a Control Data 6600), which was already burdened with supporting the real-time, piloted simulations.

The resulting program, the adaptive maneuvering logic (AML, refs. 6 and 7), distinguished itself as a formidable adversary against human pilots. In a real-time simulation with F-4 aircraft, the AML was able to consistently beat experienced pilots in 1v1 air combat maneuvering (ACM). In fact, the real-time performance of the AML is so impressive that it is used successfully as a training tool in several military simulation facilities. However, to achieve real-time performance on the 1960's vintage computer equipment in use in the DMS at the time, the AML has several key limitations that have curtailed its use except as an invariant opponent. These limitations have led to the development of TiGRES.

Three factors severely degrade the suitability of the AML simulation environment for use as a research tool. First, the motion of the aircraft is described by a five-degree-of-freedom (d.o.f.) "performance model," rather than a more standard approach with six d.o.f. As described in reference 8, the basic idea of the performance model is to move the aircraft in a realistic-appearing manner during the transition from the current aircraft attitude to one that corresponds to a commanded or desired flight condition. In this performance model, no moments are calculated; therefore, no rotational differential equations of motion are used to model the rotational dynamics of the aircraft. Instead, body-axis rotation rates (p , q , and r) are calculated directly as required to make the transition from the current body-axis attitude (defined by the Euler angles ψ , θ , and ϕ) to the commanded attitude. The required rates are approximated through the following relations:

$$\left. \begin{aligned} p &= (\Delta\phi - \Delta\psi \sin\theta) / \Delta t \\ q &= (\Delta\theta \cos\phi + \Delta\psi \cos\theta \sin\phi) / \Delta t \\ r &= (\Delta\psi \cos\theta \cos\phi - \Delta\theta \sin\phi) / \Delta t \end{aligned} \right\} \quad (1)$$

where

$$\Delta\psi = \psi_{\text{com}} - \psi_{\text{cur}}$$

$$\Delta\theta = \theta_{\text{com}} - \theta_{\text{cur}}$$

$$\Delta\phi = \phi_{\text{com}} - \phi_{\text{cur}}$$

$$\Delta t = \text{Time increment of simulation}$$

and the subscripts **com** and **cur** refer to command and current. To prevent the aircraft from rotating at unrealistic rates, limits are placed on the maximum allowable p , q , and r . If the required p , q , or r as calculated from equation (1) exceeds a maximum allowable value, that value is used instead of the calculated value. The number of d.o.f. of this performance model is five rather than six because the aircraft is always assumed to be in an attitude without sideslip, hence removing one d.o.f.

The performance model greatly reduces the computation time and data storage required to simulate a given aircraft. The performance model also significantly simplifies the task of tracking commanded trajectories. These trajectories are characterized by a desired load factor n and a maneuver-plane rotation angle ρ_m , which is defined as the angle from the negative gravitational vertical axis $-Z_E$ (i.e., upward) to the "maneuver plane" of the aircraft. This plane is defined by the velocity vector of the aircraft and the net force vector (i.e., vector sum of the gravitational, aerodynamic, and thrust forces) affecting the aircraft. Because by definition no unbalanced forces are affecting the aircraft outside the maneuver plane, the maneuver plane contains the trajectory of the aircraft. The desired n and ρ_m can be converted into a corresponding body orientation for the current flight condition. Because the performance model allows the body rotation rates to be commanded directly, the commanded trajectory is easily captured and tracked. The motion is adequate for use as an invariant opponent because, from the perspective of a pilot flying against it in a simulator, the motion does appear "realistic." However, to be a useful tool for performing analyses, the motion must be realistic in a physical sense rather than just appearing realistic. Close-in ACM engagements consist almost entirely of transient maneuvering, and failure to model the dynamics of the aircraft accurately during this maneuvering will yield incomplete results.

An interesting note is that the original developers of AML were well aware of the limitations of the performance model. When a sufficiently powerful computer (a Control Data Cyber 175) became available in the DMS to handle a six-d.o.f. model, such a model was developed and compared with both the performance model and pilots. (See refs. 7 and 9.) The results of these tests showed that, although the overall combat performance of the two models was similar, significant differences existed between the types of maneuvers performed by the performance model and by the six-d.o.f. model. However, because the primary interest in AML was still on providing an invariant opponent, the similar combat performance

of the two models was taken as validation of the suitability of the performance model in this capacity. After these tests were completed, no further work appears to have been done with the six-d.o.f. model.

The second deficiency of the AML simulation environment is that it provides only for 1v1 air combat simulation. Although 1v1 investigations are very useful for preliminary analysis, complications (e.g., cooperative tactics) of air combat that involves multiple aircraft (three or more) make multi-aircraft simulations necessary to fully investigate and understand the effect of a given concept. The reformulation from an existing 1v1 simulation to a multi-aircraft capability would not normally be prohibitively difficult. However, the lack of organization and documentation makes this upgrade less attractive in the case of AML, as described in the following paragraph.

The final deficiency of the AML simulation environment is that the FORTRAN code that implements the equations of motion was done in an ad hoc manner with various undocumented alterations and experiments scattered about. For example, elements of the maneuver decision process are implemented in the equations of motion routine simply because they were easier to implement there and may have increased execution speed. Having parts of the maneuver decision process scattered around in the simulation routines not only makes following the equations of motion more difficult, it makes tracking the decision process nearly impossible. Aircraft may perform maneuvers in a manner that is inconsistent with the intended decision process because remnants of an earlier decision logic were “hard wired” into the code that implements the equations of motion. This convoluted code is extremely difficult to upgrade reliably.

Thus, based on the need to provide a more realistic air combat simulation along with the difficulty of upgrading the simulation environment of AML to meet this need, the decision was made to develop the TMS as a new program. The experience gained from working with the AML has been helpful in defining a set of objectives for the TMS. To support the research objectives of TiGRES, TMS requires the following features:

1. The aircraft simulation model must be functionally equivalent to models used for piloted simulation studies in the DMS. This equality will allow a common tactical decision generator (TDG) to be tested against baseline decision logics in batch simulations and against pilots in the DMS. Any differences between batch and piloted simulation

results will be directly attributable to differences in maneuver strategies.

2. Current TDG's use n and ρ_m to characterize the desired trajectory. The performance model used by the AML allows the corresponding lift coefficients C_L and ϕ to be commanded directly. Unfortunately, a model that is equivalent to a piloted simulation model mandates the use of six-d.o.f. dynamics. With these higher order dynamics, the ability to command lift and bank angle directly is lost. A control system or autopilot must be added to the aircraft model to issue commands to the inner-loop control system so that the aircraft can capture and track the desired trajectory in near-minimum time.
3. The TMS must support simulation of multiple aircraft. The DMS currently has hardware to simulate and project three aircraft, which limits tests in this facility to 1v2 scenarios. However, because future upgrades to the DMS can be anticipated, the structure of the TMS should accommodate MvN participants.
4. The TMS must function as an independent element, with the information flow between the TMS and the TDG handled in a structured and easily controlled fashion. This separation is intended to prevent functions of TDG's from being inadvertently implemented in the TMS.

As will be shown in the following sections, the simulation environment described in this report meets these objectives.

Tactical Maneuvering Simulator

Functional Overview

The TMS provides a batch simulation environment for developing and evaluating tactical maneuvering strategies. The TDG's that implement various maneuvering strategies are tested against one another in varying initial conditions. The resulting trajectories can then be used to refine these strategies. Multiple iterations through this refinement process permit a globally effective maneuver strategy to be developed for a given aircraft. The TMS can also be used to evaluate the tactical implications of perturbations to aircraft performance or supporting systems. By comparing the combat performance of a modified aircraft (and appropriate TDG) with a baseline aircraft, designers can assess the effect of the modification. This assessment should provide an indication of the overall value of that modification in terms of an exchange ratio and the types of tactical maneuvers and situations that favor the modified aircraft.

The TMS provides an air combat environment with any number of engagement participants. A parallel implementation structure allows individual aircraft simulations to be initiated or "spawned" as needed. The number of aircraft being simulated at one time is limited only by the available computer memory and the desired computation speed of the simulation. Equations of motion for six d.o.f. are used to model the motion of each aircraft and data representative of a high-performance aircraft both with and without TV systems are available for use in these equations. The user is thus able to compare the performance of an enhanced agility, TV aircraft with that of an aircraft of conventional agility. The equations and data used to model the aircraft in the TMS are also implemented for piloted simulations in the DMS. This implementation provides the desired commonality between the batch and piloted simulation environments of TiGRES.

The TMS has three basic elements. The first element is the aircraft simulation model, which simulates the motions of each participating aircraft. The second element is the tactical autopilot (TA), which controls the aircraft such that it captures and tracks the trajectory commanded by its corresponding TDG. The third element is the TMS executive program, which enables multi-aircraft simulation by spawning individual aircraft, as needed, by overseeing the engagement in a common inertial reference frame and by controlling communication between aircraft and TDG's. These elements are described in the following three sections.

Aircraft Simulation Model

Individual aircraft are modeled with a modified version of an existing batch simulation model developed at the Langley Research Center. This simulation models an F-18 aircraft with or without a hypo-

thetical, hardware-based TV system developed by the Northrop Corporation. This TV system uses two vectoring vanes on each engine to provide thrust-induced pitching and yawing moments. To distinguish between the aircraft equipped with the TV system and the basic aircraft, the basic aircraft is referred to as the baseline aircraft, whereas the aircraft with the TV system is referred to as the TV aircraft. The batch simulation was developed from the real-time simulation code for the F-18 aircraft as implemented in the DMS and from documentation obtained from the McDonnell Aircraft Company. An in-depth description of the batch simulation has been published (ref. 10), but details relevant to use in the TMS are presented here.

Implementation of simulation. The computer code that implements the simulation model is written in the advanced continuous simulation language (ACSL) (ref. 11) and FORTRAN. (See ref. 12.) The ACSL is a simulation system with a special-purpose high-level language, a translator, and various libraries to satisfy the commands available in the language. The ACSL simulation models are translated into FORTRAN and linked with the ACSL libraries. The resulting executable program allows interactive user input and enables the generation of plots and printed outputs. The ACSL allows FORTRAN subroutines to be integrated into the simulation model.

The simulation uses the ACSL to implement the dynamics of the aircraft and engines. Actuator and sensor models are also implemented in the ACSL. FORTRAN subroutines are used to calculate aerodynamic forces and moments and steady-state engine parameters. The discrete, inner-loop, control augmentation system of the aircraft is also implemented primarily in FORTRAN.

Equations of motion. The equations of motion in the ACSL simulation effectively model the flight of a rigid airplane over a flat, nonrotating Earth. The aircraft mass and moments of inertia are set at the start of a simulation and are assumed to be constant. The aircraft is considered to be symmetric about the plane defined by the X and Z body axes, so that the I_{XY} and I_{YZ} products of inertia are zero and are not included in the equations. With these simplifications the equations take the following form:

Translational equation

$$m \frac{d}{dt} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} + m \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = mg \begin{Bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{Bmatrix} + \begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix}_A + \begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix}_E \quad (2)$$

Rotational equation

$$\begin{bmatrix} I_X & 0 & -I_{XZ} \\ 0 & I_Y & 0 \\ -I_{XZ} & 0 & I_Z \end{bmatrix} \frac{d}{dt} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} = \begin{Bmatrix} M_X \\ M_Y \\ M_Z \end{Bmatrix}_A + \begin{Bmatrix} M_X \\ M_Y \\ M_Z \end{Bmatrix}_E \quad (3)$$

Kinematic relations

$$\frac{d}{dt} \begin{Bmatrix} \phi \\ \theta \\ \psi \end{Bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} \quad (4)$$

Typical weights and moments of inertia used for the baseline and TV aircraft are shown in table I. Aerodynamic and thrust-induced forces and moments are discussed below.

Aerodynamic forces and moments. The aerodynamic characteristics of the simulated aircraft are discussed in detail in references 13 and 14. Figure 1 illustrates the configuration of the aerodynamic surfaces and controls. Table II provides dimensional data relevant to these aerodynamic effectors. The aerodynamic force and moment generated by each surface or control are calculated from a large wind-tunnel-derived data base using table look-ups with linear interpolations. Data are stored in a non-dimensional form as functions of angle of attack α , angle of sideslip β , Mach number M , the time rates of change of α and β , surface deflections, and rates p , q , r . The α range is -10° to 90° , the β range is -20° to 20° , and the M range is 0.20 to 2.00. Flexibility effects in the form of flex-rigid ratios and flexibility increments are included in the data base to an altitude of 60 000 ft. Actuators for all control surfaces except the speedbrake are modeled with a first-order lag with time constants and rate limiting, as in table II. The actuator responsible for moving the speedbrake is modeled as producing a constant deflection rate of 24 deg/sec.

Engine forces and moments. Two engines rated at 16 100 lb of installed static sea level thrust are included in the simulated aircraft. The engine model takes inputs from the throttle and current air data (altitude h , dynamic pressure \bar{q} , and M) to compute the force produced by the engines. For the TV aircraft, α and β effects as well as thrust losses attributable to vectoring are included in the thrust computation. Given this information, the body-axis components of thrust for each engine are computed as

$$\left. \begin{aligned} TH_{X_R} &= th_R \cos(\text{elev}_0 + \delta_{\text{elevTV},R}) \cos(\text{azim}_0 + \delta_{\text{azimTV},R}) \\ TH_{X_L} &= th_L \cos(\text{elev}_0 + \delta_{\text{elevTV},L}) \cos(\text{azim}_0 + \delta_{\text{azimTV},L}) \\ TH_{Y_R} &= th_R \cos(\text{elev}_0 + \delta_{\text{elevTV},R}) \sin(\text{azim}_0 + \delta_{\text{azimTV},R}) \\ TH_{Y_L} &= -th_L \cos(\text{elev}_0 + \delta_{\text{elevTV},L}) \sin(\text{azim}_0 + \delta_{\text{azimTV},L}) \\ TH_{Z_R} &= th_R \sin(\text{elev}_0 + \delta_{\text{elevTV},R}) \sin(\text{azim}_0 + \delta_{\text{azimTV},R}) \\ TH_{Z_L} &= th_L \sin(\text{elev}_0 + \delta_{\text{elevTV},L}) \sin(\text{azim}_0 + \delta_{\text{azimTV},L}) \end{aligned} \right\} \quad (5)$$

The elevation angle of the engine is defined in the aircraft X - Z plane; positive direction is the thrust directed in a positive Z -direction. The azimuth angle is measured in the aircraft plane; positive direction is thrust directed inward toward the vehicle centerline. For the baseline aircraft, the elevation angle is 0° , azimuth angle is 1.98° , and the δ_{TV} terms are 0. The TV aircraft is equipped with a TV system that has two vanes per engine as shown in figure 2. The change in elevation and azimuth angle produced by the TV system is defined by

$$\left. \begin{aligned} \delta_{\text{elevTV}} &= \sin^{-1}(\sin 48^\circ \sin \delta_{th}) \\ \delta_{\text{azimTV}} &= \sin^{-1}(\cos 48^\circ \sin \delta_{th}) \end{aligned} \right\} \quad (6)$$

where δ_{th} is the thrust deflection angle in degrees.

By deflecting the thrust of the two engines in a symmetric or nonsymmetric manner, a researcher can generate nearly pure pitching or yawing moments that are similar to those of an aerodynamic V-tail aircraft.

The actuators for the TV vanes are modeled as first-order transfer functions with a steady-state gain of one, a time constant of 1/30 sec, rate limits of 80 deg/sec, and position limits of $\pm 30^\circ$.

The force and moment terms in the equations of motion can now be computed as

$$\left. \begin{aligned} F_{X_E} &= TH_{X,L} + TH_{X,R} \\ F_{Y_E} &= TH_{Y,L} + TH_{Y,R} \\ F_{Z_E} &= TH_{Z,L} + TH_{Z,R} \\ M_{X_E} &= TH_{Z,R}Y_{\text{eng}} - TH_{Z,L}Y_{\text{eng}} - F_{Y_E}Z_{\text{eng}} \\ M_{Y_E} &= -F_{Z_E}X_{\text{eng}} + F_{X_E}Z_{\text{eng}} \\ M_{Z_E} &= F_{Y_E}X_{\text{eng}} + TH_{X,L}Y_{\text{eng}} - TH_{X,R}Y_{\text{eng}} \end{aligned} \right\} \quad (7)$$

Control augmentation system. As a fly-by-wire aircraft with a full authority control augmentation system (CAS), the dynamic characteristics of the simulated aircraft depend heavily on the actions of this CAS in addition to the underlying open-loop dynamics described above. This CAS is documented in detail for the baseline aircraft in references 15 and 16. A simulation of the “auto flap up” mode of the CAS defined by the version 8.3.3 production programmable read-only memory (PROM) set is in the simulation model. This auto-flap-up operational mode of the CAS is normally engaged during ACM. The CAS can be divided into control loops about the longitudinal, lateral, and directional axes. The longitudinal CAS and the other two controllers have minimal coupling; however, the lateral and directional controllers are coupled through various interconnections and will be described together.

The longitudinal CAS, shown in figure 3, uses the longitudinal stick position as the command input. The forward path gains are air data scheduled to yield a uniform initial pitch acceleration response for sharp stick inputs. A forward loop integrator drives to zero the steady-state error between the maneuver command (from longitudinal stick position) and the feedback variables. The CAS feedback is an air data scheduled blend of pitch rate, normal acceleration, and angle of attack. Pitch rate and normal acceleration feedbacks give improved pitch dynamic characteristics and load factor control in the mid- to high-dynamic-pressure portion of the flight envelope. Improved ACM flying qualities and increased stick-force-per- g cues in the low- to mid-dynamic-pressure flight regime are provided by the air-data-scheduled pitch rate feedback. Angle-of-attack feedback provides additional increased stick force cues for low-

speed, high- α ACM. Roll rate multiplied by yaw rate is fed to the longitudinal CAS to reduce the effects of inertial coupling. The longitudinal CAS also schedules the deflection of the leading- and trailing-edge maneuvering flaps as a function of α and air data to optimize performance, improve high- α characteristics, and provide load alleviation at elevated load factors.

The lateral and directional CAS, shown in figure 4, sums lateral stick position with roll rate feedback to provide closed-loop control of the ailerons, differential stabilators, differential trailing-edge flaps, and differential leading-edge flaps. The lateral CAS command path consists of structural notch filters and air-data-scheduled gains. The gains vary with \bar{q} , static pressure, and α to provide acceptable loop stability and roll response characteristics throughout the flight envelope. Maximum roll rate is limited to 220 deg/sec when normal loads are less than $5g$ and 150 deg/sec for normal loads greater than $5g$. The directional CAS uses a command signal from the rudder pedals with stability-axis yaw rate ($r \cos \alpha - p \sin \alpha$) and lateral acceleration feedback. The rudder pedal force transducer signal is α and air data scheduled to prevent a command that would exceed the vertical tail load limits and to eliminate aircraft departures for full pedal inputs. The $r \cos \alpha$ feedback component helps provide sideslip reduction during moderate and high- α maneuvering flight. Lateral acceleration feedback aids in reducing sideslip and provides turn coordination. Roll rate multiplied by pitch rate is fed to the directional CAS to reduce the effects of inertial coupling. The lateral and directional controllers are coupled through a rolling-surface-to-rudder interconnect and a rudder-pedal-to-rolling-surface interconnect. The

rolling-surface-to-rudder interconnect is incorporated to minimize sideslip that could accompany lateral stick inputs. Similarly, the rudder-pedal-to-rolling-surface interconnect is provided to reduce sideslip and α excursions from rudder pedal inputs at high α . The interconnect is scheduled with α and is scheduled to zero at low α .

The CAS used with the TV aircraft is a refined and extended version of the baseline CAS. This work was performed by the Flight Dynamics Branch at the Langley Research Center through extensive batch and piloted simulation analyses. The CAS integrates the TV system with the aerodynamic control surfaces to significantly increase the maneuvering capabilities of the aircraft at high α . The feedback structure and operation of this CAS are similar to those described for the baseline aircraft. The pitch and yaw commands from the command paths are divided, as appropriate, between the aerodynamic and TV controls. The pitch and yaw commands sent to the TV system are passed through a mixer that resolves the commands into appropriate vane deflection commands for the TV hardware of the left and right engines.

The CAS described above augments the dynamics of the bare airframe to provide stability and predictable flying qualities that enable pilots to employ the aircraft in tactical engagements. For use in the TMS, an outer-loop control system is needed around the basic CAS to track trajectories commanded by a TDG. In a sense, this outer-loop control system performs the physical functions of the pilot—that is, it transforms the desired tactical plan or strategy into actual aircraft motions. This outer-loop control system, the TA, is described in the following section.

Tactical Autopilot

The TA accepts trajectory commands generated by a TDG and issues commands to the inner-loop CAS that cause the aircraft to follow the desired trajectory. Current TDG's issue trajectory commands by specifying parameters that define a desired magnitude and orientation for the lift force combined with a desired throttle and speedbrake setting. Because the throttle and speedbrake settings are obtained directly, no interface is needed to capture these commands; the commands are passed directly from the TDG to the aircraft simulation. In contrast, the magnitude and orientation of the desired lift force cannot be obtained directly, which requires the development of the TA.

Many different parameter pairs can be used to specify the desired lift vector. For a given

flight condition, the magnitude of the lift vector can be specified by commands to the corresponding desired load factor to C_L or to α . Similarly, the orientation can be specified by various angular references such as ρ_m , ϕ , or wind-axis bank angle μ , which is defined as

$$\mu = \tan^{-1} \left(\frac{\sin \theta \cos \alpha \sin \beta + \sin \phi \cos \theta \cos \beta - \cos \phi \cos \theta \sin \alpha \sin \beta}{\sin \theta \sin \alpha + \cos \phi \cos \theta \cos \alpha} \right) \quad (8)$$

Equation (8) is obtained from the matrices that transform vectors from Earth axis to body axis \mathbf{L}_{BE} (ref. 17) and body axis to wind axis \mathbf{L}_{WB} (ref. 17) to calculate Earth axis to wind axis \mathbf{L}_{WE} (ref. 18) with the relationship $\mathbf{L}_{WE} = \mathbf{L}_{WB} \mathbf{L}_{BE}$.

For modern, high-performance aircraft, specification of α and μ offers several advantages. First, to fully exploit the tactical potential of these advanced aircraft, the TDG must command maneuvers in the stall/poststall region. During those maneuvers, the aircraft orientation is frequently more important than its flight path. Because lift curve slopes are generally shallow and variable in the stall/poststall region, orientation relative to velocity vector is poorly defined by load factor and C_L . In contrast, α remains an effective command variable in the stall/poststall region. Second, an awareness of α is ensured in the TDG. Because the current and future maneuvering potential of an aircraft is largely a function of α , this awareness is imperative to the formulation of effective maneuver decisions and strategies. Third, μ directly specifies the desired orientation of the lift vector, thereby eliminating the need to calculate the corresponding body-axis bank angle while ensuring that the vector is oriented as intended.

The TA thus is an all-attitude, outer-loop control system to capture and track α and μ as commanded by a TDG. Coordinated flight (defined as flight with $\beta = 0$) is assumed desirable at all times. A block diagram of the complete TDG TA aircraft system is shown in figures 5 and 6. The TA described in this paper represents an initial design and allows current TDG's, intended to operate with five-d.o.f. performance models, to interface with and effectively command full six-d.o.f. models. The TA enables this interface with minimal modifications to these existing TDG's. As experience is gained from these initial efforts, the design of the TA can be refined as performance requirements and even desired command variables become better defined. For instance, full exploitation of the nose-pointing capability of the simulated aircraft may make $\beta = 0$ not desired at all times.

The task performed by the TA is similar to the function of the control system developed for the six-d.o.f. model test in the AML. This control system, which is described in reference 9, allowed the guidance logic of the AML to effectively command a six-d.o.f. simulation of an F-4 aircraft. Because of this success and the similarity to the current application, reference 9 has been a guide during the development of the TA. The design and development of the TA is described in detail in reference 19 and is summarized herein. Although the TA is described in this report in the context of the TMS, its use is also required in the DMS. The incorporation of the TA into the piloted simulation model of the DMS permits the TDG's to command this simulation in an identical manner to the batch simulation.

The TA is divided into two channels – a longitudinal command system that uses longitudinal stick inputs to capture and track commanded α and a lateral command system that uses lateral stick inputs to capture and track the commanded μ . A directional controller is not included in the TA because the inner-loop CAS already attempts to maintain zero sideslip, unless commanded otherwise by the rudder pedal inputs. Piloted simulations have shown that the wind-axis rolling performance of the baseline aircraft can be improved slightly at $\alpha > 25^\circ$ by rudder pedal inputs. (See ref. 20.) This performance is not being exploited by the current TA.

The longitudinal command system uses a proportional-integral-derivative (PID) structure with α feedback, as shown in figure 6(a). The lateral command system uses a proportional-derivative (PD) structure with μ feedback, as shown in figure 6(b). The values of α , the rate of change of α ($\dot{\alpha}$), μ , and the rate of change of μ ($\dot{\mu}$) are assumed to be available without error, so no additional compensation to account for sensor noise or dynamics is included in the TA. Also, no attempt is made to model the cognitive and neuromuscular delays or limitations that are inherent in a human pilot. Thus, as implemented, the TA represents an idealized controller.

The gains for the command systems were determined through a combination of linear analysis and evaluation of the full nonlinear system response to step commands and representative command sequences. To obtain good performance throughout the ACM envelope of the simulated aircraft, the three gains of the longitudinal command system ($K_{P\alpha}$, $K_{D\alpha}$, and $K_{I\alpha}$) are scheduled as a function of \bar{q} . In addition, $K_{D\alpha}$ is also scheduled as a function of density ratio ρ/ρ_0 to compensate for changes in aerodynamic damping with altitude. Good performance across the ACM envelope is achieved by the lateral

command system by the scheduling of its two gains $K_{P\mu}$ and $K_{D\mu}$ with α .

To achieve time-optimal control of a system with limited control authority, generally the maximum available control authority must be used at all times. (See ref. 21.) Because the TA should capture commands in minimal or near-minimal time, the gains of the command systems have been selected such that the commanded stick positions are frequently near saturation for small command changes and saturated for moderate and large changes. This saturation does not cause significant difficulties for the lateral command system. Gains $K_{P\mu}$ and $K_{D\mu}$ are selected such that the lateral stick input becomes unsaturated with sufficient control authority remaining for the linear controller to capture the desired μ with acceptable levels of overshoot. Saturation can cause problems with the longitudinal control system unless the action of the integral element is restricted to prevent integrator windup. If the gain on the integral element is adjusted such that good response is achieved for small command changes, large overshoots are obtained for moderate and large changes. During these changes, the maximum rate is quickly reached at which α can be increased (or decreased). Because of this nonlinear, rate-limited performance, the longitudinal stick command from the integral control action can reach very high levels during the initial response. The integral of the α error decreases only after the desired α is exceeded, so large overshoots can result. To prevent this windup, the calculation of the integral of the α error is suspended when the sum of the longitudinal stick commands from the proportional and rate elements causes saturation. This suspension is bypassed if the current integral command is in opposition to the direction of saturation. This bypass is necessary to efficiently respond to command changes that involve a sign change in α error.

During evaluations of system response to coupled, large-amplitude α and μ commands, the baseline aircraft was discovered to be prone to departures at relatively low α when full or nearly full lateral stick inputs are used and when the longitudinal stick input is aggressively increased to maintain constant α . As shown in figure 7, the departure results because β builds to excessively high levels as the rudders saturate against their deflection limits. This departure results when the inner-loop CAS allows the aircraft to obtain a roll rate beyond its ability to remain coordinated. As the departure represents a potentially dangerous flight characteristic, the phenomenon was investigated further in piloted simulation with the DMS. A similar, but less violent response was reproduced in the piloted simulation.

The β departure occurred only after the aircraft had rolled through 360° . In tactical maneuvering, full lateral stick will not likely be maintained much beyond a 180° roll; thus, this performance is unlikely during normal operations. The difference in departure characteristics observed in the TMS and DMS may be caused by the abrupt control commands issued by the TA versus those of a human pilot. To prevent the baseline aircraft from departing while under the control of the TA, the allowable stick input must be limited in the affected α range. For $\alpha < 15^\circ$, the input is limited to 85 percent of the maximum lateral stick travel. For $\alpha > 15^\circ$, the limit is relaxed in a linear fashion until full travel is available at $\alpha = 20^\circ$.

One difficulty in developing a system such as the TA is the determination of suitable criteria with which to measure the acceptability of the final design. Traditional performance specifications such as frequency and damping are inappropriate because of the large-amplitude, coupled maneuvers performed by the TA. Criteria that reflect the nonlinearities of the task must be used to assess TA performance. The intent of these criteria is to ensure that the TA can capture and track commands from the TDG without adversely biasing the tactical performance of the TDG TA aircraft system. This tactical performance is dependent on the combined interactions of all three components, so the response of the TA aircraft system should be characterized in relation to some functional benchmark. Because the only previous controllers to demonstrate mastery of the simulated aircraft in ACM are human pilots, the performance of pilots with representative maneuvers can provide a benchmark for TA performance.

Tables III and IV show the minimum and average time required for a series of experienced pilots to perform large-amplitude, decoupled α and μ captures in the baseline and TV aircraft, as simulated in the DMS. Also shown in the tables is the time required by the TA to perform the same captures. Time histories for these TA maneuvers are presented in figures 8 and 9. All runs start from $1g$ level flight and end when the desired α or μ is captured within the specified tolerance. The tables show that for all but two of the tasks, the TA required less time than did the pilots. The TA is probably able to consistently perform the desired maneuvers in less time than the human pilots because it can respond instantly to the current situation. In the two tasks in which the TA did not outperform the pilots, the performance differences are small.

For the 90° roll maneuver at $\alpha = 10^\circ$ with the TV aircraft, the TA takes 0.06 sec longer than the minimum piloted time. This increase is probably

tactically insignificant and may be attributable to α variations during the maneuver. Data recorded during the maneuver show that the pilot allowed the α to fall to 7.2° during the maneuver; the TA minimum α was 8.5° .

For the capture task at $M = 0.60$ and $\alpha = 40^\circ$ with the baseline aircraft, the TA was unable to prevent the initial overshoot from exceeding the desired $\pm 2.0^\circ$ capture tolerance. This overshoot increased the capture time of the TA for the original capture tolerance beyond the minimum piloted time. The initial TA overshoot was 0.44° beyond the desired capture tolerance. As this overshoot only slightly exceeds the desired capture tolerance, the tactical performance should not be significantly affected. Because attempts to improve the response at this one condition resulted in an overall decrease in system performance, the decision was made to accept the nominal response of the system. The time listed in table III represents the performance of the TA with the capture criteria relaxed to 2.44° .

Also shown in the table is the maximum peak overshoot M_P for the TA captures. Burgin and Eggleston (ref. 9) recommend that for good tactical performance, M_P for decoupled inputs should be limited to 5° in pitch and 20° in roll, regardless of the amplitude of the input. For all the captures, the TA is below these recommended limits.

The capture tasks shown in tables III and IV measure performance for single-axis, step inputs. In ACM, the TA will be expected to respond to sequences of simultaneous α and μ commands. The responses of the TA to a representative command sequence are shown in figures 10 and 11 for the baseline and TV aircraft, respectively. These command sequences were obtained by discretizing, at 1-sec intervals, continuous α and μ time histories recorded during piloted ACM engagements. This discretization was performed to obtain command sequences that are representative of the command update rate of a typical TDG. Because these command sequences were obtained from actual trajectories, the sequences should be reasonably close to the capabilities of the TA-controlled aircraft and representative of a tactically realistic command sequence.

The TA appears to follow both sequences with sufficient accuracy to effectively implement realistic maneuver sequences. As shown in figures 10 and 11, the ability of the TA to capture and maintain α and μ is only slightly reduced by the coupled command sequences. However, an absolute, operational assessment of TA effectiveness cannot be performed until

the system is interfaced with an appropriate TDG and tested against human pilots in the DMS.

Multiple Aircraft Simulation and TMS Executive Program

The TMS uses a novel parallel implementation technique to provide multi-aircraft simulations. Most batch multi-aircraft simulation environments are implemented as a single large process. A main program calls various subroutines to implement the engagement participants. The researcher can create additional participants by duplicating the requisite subroutines, by renaming variables and common blocks as necessary to avoid memory conflicts, and by updating the calling sequence of the program. The TMS exploits parallel processing libraries provided by the Digital Equipment Corporation VAX/VMS 5.0® operating system (ref. 22) to implement simulation participants as independent processes that communicate with and are synchronized by a master process through a shared block of memory. This implementation allows a single copy of the simulation program to run concurrently as needed to simulate the individual engagement participants. Because they are run as independent processes, memory conflicts are avoided without the need to manually modify each participant. The number of concurrent copies of the simulation that can be executed simultaneously is limited only by available computer memory and the desired execution speed of the simulation. Of course, an appropriate TDG would be needed to command the aircraft.

In addition to simplifying the simulation of multiple aircraft, this parallel implementation offers several other key advantages compared with conventional methods. Because all aircraft are simulated by the same program, corrections or updates to this model need only be performed once, which eases configuration control issues. With a conventional implementation, these changes must be repeated in each duplicated subroutine. This need to repeat changes is frequently a source of difficulty, as the odds of a programming error increase with each repetition. As will be shown, the current parallel implementation allows different simulation models to be incorporated into the TMS and be intermixed with the current aircraft simulation model with only the addition of a standard subroutine. Thus, simulations of different aircraft types can easily be added to the TMS, which allows comparisons of the tactical performance of different types of aircraft. Simulations that may be added to the TMS are not restricted to aircraft; for example, high-fidelity missile simulations could also be implemented in a similar fashion. Finally, al-

though not investigated in this study, parallel implementation should allow individual simulations to be distributed on multiple, networked computers. Thus, if the number of simulation participants grows beyond the capacity of a single computer, the ability to use distributed processing on an existing computer network may obviate the need to purchase a more powerful computer.

The concurrent parallel implementation provides the above-mentioned benefits, but a control mechanism is needed to synchronize the otherwise independently executing simulations. This synchronization is required so that the simulations remain together on the same time step. Because the simulations execute as independent processes on a given computer (or computers), the order and length of time in which the computer operates on each process are functions of other jobs on the machine and are essentially indeterminate. Thus, without some type of control mechanism, the simulations would progress at different rates.

The TMS uses barrier synchronization to control the progress of individual simulations. Barrier synchronization involves the use of barrier statements that suspend execution of individual processes at a specified point until all relevant processes have reached their respective barriers. After all processes have reached the barrier statements, the processes are allowed to continue execution. Barriers are used in the TMS to suspend the execution of the aircraft simulations at the end of the current time step or simulation frame. The simulations are allowed to proceed only after all simulations have reached the end of the current time step.

The key elements of the parallel implementation used by the TMS are a FORTRAN executive program and a FORTRAN subroutine that was added to the aircraft simulation model to communicate with the executive program and to enable the executive to synchronize the concurrently executing simulations. The executive program is a master process that initializes the individual simulation models and supervises their operation. The executive program also handles communication with the TDG's and passes information to and from the TDG's by means of subroutine calls. Because all communication between a TDG and its corresponding aircraft must pass through the executive program, the flow of information can be closely monitored and controlled. The final function of the executive program is to track and "score" the engagement in a common reference frame. The executive program uses data returned from the simulations to determine the current relative geometry between aircraft. These

relative geometries are used to score the engagement by calculating the probability that each aircraft will successfully fire a weapon at opposing aircraft. This probability of kill P_k is currently based on very simple models of the firing envelopes of a modern, all-aspect, air-to-air missile and a high-velocity gun. The operational interaction between the executive program, the aircraft simulation model, and the TDG is shown graphically in figure 12 and is described below.

The TMS executive program consists of two primary sections of code. The first section is presented in simplified form in appendix A and sets up the area of shared memory used to communicate with the other processes. This memory is contained in the common block `SHARED.DATA`. This common block is analogous to a standard FORTRAN common block, but rather than being shared among subroutines of a single process, this common block can be shared by concurrently executing processes. Next, a do-loop is used to initialize each simulation participant. The command files executed by the `LIB$SPAWN` command assign unique input and output files to each aircraft simulation. Each time the `PPL$SPAWN` command is performed, the executable code of the simulation model (`F18XX.EXE`) is initialized as a new process. The command `PPL$WAIT.AT.BARRIER(BARRIER.INT)` keeps the simulation from proceeding prematurely and causing difficulties during the assignment of input and output files. A corresponding barrier is in the initialization code of the simulation model. At the completion of this first section of code, the simulations have been initialized and are waiting to continue execution at time zero.

The second section of the executive program maintains the synchronization of the simulations, scores the engagements, and calls the TDG's at each time step. This second section of code interacts with the previously mentioned communication and synchronization subroutine. This subroutine, shown in appendix B, is implemented in the simulation model as the last routine to be executed. Just before the individual simulations reach the barrier `BARRIER.DATA`, the data shared with the executive program are updated to the current time step. These data include the current attitude, position, velocity, rotation rates, control positions, and thrust of the aircraft. The data from a specific aircraft can be identified by `MY INDEX`. As each process is spawned, the operating system assigns it a unique integer index that can be retrieved by the command `PPL$GET INDEX`. After all the simulations have reached `BARRIER.DATA`, the executive

program is allowed to proceed to the relative geometry and P_k calculations and to communicate with the TDG's. The TDG's return updated maneuver commands in the form of desired α , μ , throttle positions, and speedbrake settings. During this interval, the simulations are held at `BARRIER.COMD`. When the executive program completes this communication and reaches `BARRIER.COMD`, the simulations are allowed to proceed and receive the updated maneuver commands through the shared common block. It is important to recognize that the communication and synchronization subroutine could be incorporated into most ACSL or FORTRAN simulations, so that many different simulations can be added and mixed in the TMS with minimal effort. Of course, because the TA is aircraft dependent, it would require retuning or redesigning to support other aircraft.

The following section demonstrates the capabilities of the TMS through two sample engagements.

Demonstration of Tactical Maneuvering Simulator

The operation of the TMS is demonstrated by two example engagements. The first example demonstrates TMS simulation and synchronization of four aircraft. The second example demonstrates a 1v1 engagement between a drone aircraft that follows a predefined command sequence and an actively guided aircraft.

Simulation of Four Aircraft

The parallel implementation in the TMS provides an efficient and flexible environment for simulating multiple aircraft. However, because a parallel implementation introduces the possibility of synchronization problems not found in serial programming, the barrier structure must be specifically tested to ensure that no unanticipated conflicts or problems occur. The following example is designed to demonstrate the simulation of four aircraft and to check for proper synchronization.

A simple maneuvering logic was developed to cause an aircraft that flies down the X_E -axis in a negative direction to perform a vertical reversal maneuver, shown in figure 13. This reversal consists of a half-loop followed by a 180° roll to return to upright, level flight. The maneuvering logic divides the reversal into four phases. In the first phase, the aircraft maintains $1g$ trimmed flight. In this example, the trim conditions are $M = 0.90$ at an altitude of 10 000 ft. The second phase of the maneuver begins when the aircraft passes over the Y_E -axis. During this second phase, α is commanded to 10°

while $\mu = 0^\circ$. The throttle is also maintained at its trimmed position during this initial pull-up. The third phase of the maneuver begins when the flight path angle γ passes through 90° . At this point in the trajectory, the actual μ flips from 0° to 180° . To maintain the aircraft in the desired pull-up, the commanded μ is also flipped to 180° . During this phase, the aircraft is flying in an inverted orientation relative to the inertial reference system. To circularize the trajectory, the commanded α is reduced to 6° and the throttle is increased to full afterburner. The third phase of the maneuver begins when γ passes back through 15° . At this point, both α and μ are commanded to 0° . These commands cause the aircraft to roll 180° from an inverted to an upright orientation relative to the inertial system. The final phase of the maneuver begins when this rolling command is completed. To resume approximately level flight, α is commanded to 3° and the throttle is reduced to just above its original trimmed position. The bank angle is commanded as needed to remove any lateral offset during the 180° roll.

The input or trim file used to provide the initial conditions for the simulated aircraft at the start of this maneuver is shown in appendix C. This trim file is read by the simulation and specifies the initial aircraft characteristics and flight condition. The file allows the user to vary inertial properties, select various modeling options, and specify the initial position and flight conditions. As shown in appendix C, the aircraft in this example is initialized with the inertial properties of the baseline aircraft and the modeling options are set to duplicate the DMS real-time simulation. The flight condition is specified as straight and level flight at $M = 0.90$ and $h = 10\,000$ ft. The initial position for the aircraft is set to $X_E = 5000$ ft, $Y_E = 0$, and $\psi = 180^\circ$.

The ability of the TMS to simulate and synchronize multiple aircraft is demonstrated by using the maneuver commands for this one aircraft to command three additional aircraft, starting in symmetry on the X_E - and Y_E -axes and converging toward the X_E, Y_E origin. As the original aircraft performs the reversal, the α , μ , and throttle commands are echoed to the new aircraft. The original aircraft is in a position that would be analogous to the flight leader of an aerobatic demonstration team calling out commands for the other team members to follow without question. The initial conditions and execution of this maneuver are such that if proper synchronization is maintained, the aircraft will simultaneously pass over the X_E, Y_E origin at the top and bottom of the reversal maneuver.

The TMS was configured to spawn four copies of the aircraft simulation. Trim files identical to the one shown in appendix C with the exception of the initial X_E , Y_E , and ψ were created for the three additional aircraft. The values of X_E , Y_E , and ψ of these trim files were set to provide the desired starting symmetry about the X_E, Y_E origin. As the original aircraft performed its reversal, its commanded α , μ , and throttle positions were passed through the TMS executive to the other three simulations. Thus, if synchronization is maintained in the TMS, the resulting trajectories should remain symmetrical about the origin and because of the geometry of the maneuver, the four aircraft should "collide" at the top and bottom of the maneuver. Figure 14 shows the trajectories of the aircraft during the maneuver from various perspectives. As can be seen from that figure, the reversals are completed with complete symmetry and expected intersections, and demonstrate that correct synchronization is maintained.

One-Versus-One Engagement

The second example engagement demonstrates a 1v1 dogfight between a drone aircraft in a pre-defined, open-loop command sequence and an aircraft actively guided by a simple TDG. The objective of this example is to demonstrate the operation of the TMS with a fully active TDG.

The TDG commands α and μ to cause the flight path of the guided aircraft to intersect a predicted future position of the drone aircraft. This predicted future position is obtained by extrapolation along a second-order curve fit to the past three recorded positions of the drone aircraft. The TDG then determines the maneuver plane and load factor required to intercept that position given the current state of the guided aircraft. The required maneuver plane and load factor are converted into a required α and μ . If the required load factor is outside the aerodynamic or structural capabilities of the aircraft, the α that corresponds to maximum available or allowable lift is commanded. In addition, if the commanded μ differs from the current μ by more than 45° and the commanded α is greater than 15° , the α command is reduced to 15° to expedite the execution of the rolling maneuver. This reduction was heuristically selected and does not necessarily reflect an optimal maneuvering strategy.

The engagement between the two aircraft is shown in figure 15 from various perspectives. The engagement starts with both aircraft trimmed in 1g level flight at $h = 10\,000$ ft and $M = 0.90$. Both aircraft start from opposite headings with a longitudinal separation of 10 000 ft and a lateral offset of

1000 ft. The drone aircraft is initially commanded to maintain $\mu = 0^\circ$ and to increase α slightly over the trim value. The throttle of the drone aircraft is advanced into the afterburner region. These commands are maintained during the first 10 sec of the engagement. After the initial merge, the guided aircraft responds by performing an oblique, pitch-back maneuver to reverse its heading back toward the drone aircraft. After this initial period, the drone is commanded to increase α to 28° and to alternate μ between $\pm 90^\circ$, switching every 10 sec. The resulting motion is a descending spiral trajectory. In response to these maneuvers, the guided aircraft reverses its heading again and effectively tracks the drone down the descending spiral. Time histories of commanded α versus actual α and commanded μ versus actual μ for the guided aircraft are shown in figure 16. These time histories demonstrate that the TA-controlled aircraft can closely track the TDG-generated guidance commands.

These two examples have demonstrated the operation of the completed TMS. The following sections describe potential future research activities and summarize the accomplishments of the current work.

Future Research Activities

Future research options include development of additional aircraft simulation models, incorporation of human physiological factors into the design of the TA, and the addition of an interactive user interface to allow the TMS to function as a tactical workstation.

Because the parallel implementation technique allows aircraft simulations to be added to the TMS with minimal effort, numerous existing simulations could be added to the environment, thereby providing the user with a catalog of aircraft types. An interesting model to include in this selection would be an unmanned aircraft, designed without the physiological and safety constraints imposed by a human pilot. A very illuminating test could be run that compares the performance of this type of aircraft, flown by a TDG, with conventional piloted aircraft. Use of this unmanned aircraft as an "automated wingman" to support conventional piloted aircraft could also be investigated.

The basis for the current TA was the assumption that the inner-loop control system of the aircraft provides desirable handling qualities. This assumption could be tested further by incorporating elements of pilot modeling into the TA. The field of pilot modeling is an attempt to quantify the controlling actions of a pilot through appropriate transfer functions.

Terms are incorporated into these transfer functions to reflect the physical capabilities and limitations of a typical pilot. Existing theory is limited largely to control of a single axis for small-amplitude tracking tasks and significant research would be required to extend this theory throughout the TA operating range. If successful, the TMS could provide an initial assessment of the combat effectiveness of preliminary or proposed aircraft designs as flown by a typical pilot in tactical engagements. This initial assessment has several advantages: it could be performed quickly, it would be inexpensive, it would reduce the need for piloted simulation, and it would allow designers to make more informed decisions during the design process.

The TMS currently depends on TDG's to generate trajectory commands for the simulated aircraft. However, the TMS could be easily modified by the addition of an interactive user interface to receive commands from human operators for some or all aircraft. The TMS could thus be used as a tactical workstation, allowing pilots and tacticians to explore maneuvering strategies in low-cost, nonreal-time simulations. The ability to bring the human element into ACM studies during the batch simulation phase should significantly reduce the time required to validate results in real-time, piloted simulations. To maintain the situational awareness necessary to develop effective maneuver strategies, these operators will need a large quantity of data, which can probably be conveyed most efficiently by a graphical interface. Ideally, this interface would allow pilots who are unfamiliar with the system to intuitively and effectively command simulated aircraft after a brief instruction period.

Concluding Remarks

The development and operation of a batch air combat simulation environment known as the tactical maneuvering simulator (TMS) have been presented. The TMS is a tool for developing and evaluating tactical maneuvering logics. The environment can also be used to evaluate the tactical implications of perturbations to aircraft performance and supporting systems.

The TMS was developed from an existing batch simulation of a modern, high-performance aircraft, with and without thrust vectoring. This batch simulation uses six-degree-of-freedom (d.o.f.) equations of motion, aerodynamics, propulsive characteristics, and control laws equivalent to those in high-fidelity piloted simulation.

An outer-loop control system, the tactical autopilot (TA), was developed to allow existing guidance

logics intended for use with a reduced-order aircraft model to command the six-d.o.f. aircraft model with minimal modification. The TA uses longitudinal and lateral stick inputs to capture angle of attack and wind-axis bank angle as commanded by the guidance logic. The performance of the TA was demonstrated by comparison of the time required for it to capture decoupled angle-of-attack and bank-angle commands with the time required by human pilots for the same commands. The TA performed as well as or better than the pilots for nearly all the commands investigated. The ability of the TA to track realistic command sequences of angle of attack and bank angle was demonstrated on sequences generated from piloted air combat simulations. The TA was shown to effectively track these representative command sequences.

To provide for the simulation of air combat with multiple participants, a parallel implementation scheme was developed from the parallel processing libraries provided by the Digital Equipment Corporation VAX/VMS 5.0[®] operating system. This parallel implementation allows the TMS to simulate air combat with any number of engagement participants; in fact, the maximum number is limited only by the available computer resources. The parallel implementation also simplifies software maintenance and allows new simulations to be easily added to the environment.

The capabilities of the TMS were demonstrated with two example engagements. The first engagement demonstrated TMS ability to simulate four aircraft; the second demonstrated TMS ability to interact with an active guidance logic.

NASA Langley Research Center
Hampton, VA 23681-0001
April 27, 1993

References

1. *Advanced Air-to-Air System Performance Evaluation Model (AASPEM) - User's Manual*. Survivability/Vulnerability Information Analysis Center, Booz-Allen & Hamilton, Inc., Nov. 1990.
2. Kerchner, R. M.; et al.: *The TAC Brawler Air Combat Simulation Analyst Manual (Revision 3.0)*. DSA Rep. #668 (Contract F49620-83-C-0142), Decision-Science Applications, Inc., Oct. 1985.
3. Goodrich, Kenneth H.; and McManus, John W.: Development of a Tactical Guidance Research and Evaluation System (TGRES). AIAA-89-3312, Aug. 1989.
4. McManus, John W.; and Goodrich, Kenneth H.: Application of Artificial Intelligence (AI) Programming Techniques to Tactical Guidance for Fighter Aircraft. AIAA-89-3525, Aug. 1989.
5. Ashworth, B. R.; and Kahlbaum, William M., Jr.: *Description and Performance of the Langley Differential Maneuvering Simulator*. NASA TN D-7304, 1973.
6. Burgin, George H.; Fogel, Lawrence J.; Phelps, J. Price: *An Adaptive Maneuvering Logic Computer Program for the Simulation of One-on-One Air-to-Air Combat—Volume I: General Description*. NASA CR-2582, 1975.
7. Hankins, Walter W., III: *Computer-Automated Opponent for Manned Air-to-Air Combat Simulations*. NASA TP-1518, 1979.
8. Burgin, George H.: *Improvements to the Adaptive Maneuvering Logic Program*. NASA CR-3985, 1986.
9. Burgin, George H.; and Eggleston, David M.: *Design of an All-Attitude Flight Control System To Execute Commanded Bank Angles and Angles of Attack*. NASA CR-145004, [1976].
10. Buttrill, Carey S.; Arbuckle, P. Douglas; and Hoffer, Keith D.: *Simulation Model of a Twin-Tail, High Performance Airplane*. NASA TM-107601, 1992.
11. *Advanced Continuous Simulation Language (ACSL): Reference Manual, Tenth ed.* Mitchell and Gauthier Associates, c.1991.
12. *VAX FORTRAN User Manual, Revised*. Order No. AA-D035E-TE, Digital Equipment Corp., c.1988.
13. Pelikan, R. J.; and Swingle, R. L.: *F/A-18 Stability and Control Data Report. Volume I: Low Angle of Attack*. MDC A7247 (Contract N00019-75-C-0424), McDonnell Aircraft Co., Aug. 31, 1981. (Revised Nov. 15, 1982.)
14. Hobbs, C. R.: *F/A-18 Stability and Control Data Report. Volume II: High Angle of Attack*. MDC A7247 (Contract N00019-75-C-0424), McDonnell Aircraft Co., Aug. 31, 1981.
15. Hess, R. K.; and Moomaw, R. F.: *F/A-18 Flight Control Electronic Set Control Laws*. MDC A4107 (Contract N00019-75-C-0424), McDonnell Aircraft Co., Mar. 19, 1976.
16. Groll, D. B.; Hess, R. K.; Hodges, W. D.; and Moomaw, R. F.: *F/A-18A Flight Control System Design Report. Volume I System Description and Theory of Operation*. MDC A7813 (Contract N00019-75-C-0424), McDonnell Douglas Co., Dec. 23, 1982.

17. Etkin, Bernard: *Dynamics of Atmospheric Flight*. John Wiley & Sons, Inc., c.1972.
18. Miele, Angelo: *Flight Mechanics. Volume I—Theory of Flight Paths*. Addison-Wesley Publ. Co., Inc., c.1962.
19. Goodrich, Kenneth H.: *Development of a Six Degree-of-Freedom Simulation Environment for Tactical Guidance Research and Evaluation*. M.S. Thesis, George Washington Univ., 1991.
20. Murphy, Patrick C.; Bailey, Melvin L.; and Ostroff, Aaron J.: *Candidate Control Design Metrics for an Agile Fighter*. NASA TM-4238, 1991.
21. Stengel, Robert F.: *Stochastic Optimal Control—Theory and Application*. John Wiley & Sons, Inc., c.1986.
22. *VMS RTL Parallel Processing (PPLS) Manual*. Order No. AA-LA74B-TE, Digital Equipment Corp., c.1990.

Appendix A

TMS Executive Program

The TMS executive program is presented here in simplified form. It is shown dimensioned for up to four aircraft. The function of this routine is to initialize engagement participants and oversee the engagement in a common reference frame.

```
PROGRAM TMS_EXEC
C
C  EXTERNAL DEFINITIONS
C
  INTEGER*4 PPL$SPAWN, LIB$SPAWN, PPL$INITIALIZE
  INTEGER*4 PPL$CREATE_BARRIER, PPL$WAIT_AT_BARRIER
  INTEGER*4 PPL$CREATE_SHARED_MEMORY, LIB$PUT_OUTPUT
C
C  LOCAL DATA
C
  INTEGER*4 LENADR(2), STATUS
  INTEGER*4 ONE_PAGE
  PARAMETER(ONE_PAGE = 512)
C
  REAL  RANGE(4,4), RANGE_RATE(4,4)
  REAL  LOS(4,4), AZIMUTH(4,4), DEVIATION(4,4), ANGLE_OFF(4,4)
  REAL  MIS_PK(4,4), GUN_PK(4,4)
C
C  DATA FOR SHARING
C
  BYTE      FRONT_GUARD(ONE_PAGE)
  INTEGER    COPIES
  REAL      AIRSPEED(4), ALPHA(4), BANKWND(4), BETA(4)
  REAL      DIRCOS(4,9), EULER(4,3), GAMMA(4)
  REAL      GLOAD(4), MCH(4), POSITION(4,6)
  REAL      QUAT(4,4), ROTRATES(4,6), SPDBRAKE(4)
  REAL      STKRUD(4,3), TIME(4), THRUST(4)
C
  REAL  COM_ALPHA(4), COM_BANK(4), COM_SPDBRK(4), COM_THRUST(4)
  BYTE  REAR_GUARD(ONE_PAGE)
C
C  PUT SHARED DATA IN TO COMMON BLOCK
C
  COMMON/SHARED_DATA/FRONT_GUARD,
1      COPIES,
1      AIRSPEED, ALPHA, BANKWND, BETA,
1      DIRCOS, EULER, GAMMA,
1      GLOAD, MCH, POSITION,
1      QUAT, ROTRATES, SPDBRAKE,
1      STKRUD, TIME, THRUST,
C
1      COM_ALPHA, COM_BANK,
1      COM_SPDBRK, COM_THRUST,
1      REAR_GUARD
```



```

C
CHARACTER*8 PLANE(4)
DATA PLANE/'@PLANE1','@PLANE2','@PLANE3','@PLANE4'/
C
C MAP SHARED ADDRESS SPACE
C
LENADR(1) = %LOC(REAR_GUARD) + ONE_PAGE - %LOC(FRONT_GUARD)
LENADR(2) = %LOC(FRONT_GUARD)
PRINT *, 'PEND LENADR', LENADR(1), LENADR(2)
STATUS = PPL$CREATE_SHARED_MEMORY(SHARED_DATA, LENADR)
PRINT *, 'PEND LENADR', LENADR(1), LENADR(2)
C
C LOOP TO CREATE AIRCRAFT
C
STATUS = PPL$CREATE_BARRIER(BARRIER_INT, 'BARRIER_INT', %REF(2))
PRINT *, 'INPUT NUMBER OF AIRCRAFT (1-4). '
READ (5,11) COPIES
11 FORMAT(I2)
DO 99 I = 1, COPIES
    IF (I.EQ.1) STATUS = LIB$SPAWN('@PLANE1')
    IF (I.EQ.2) STATUS = LIB$SPAWN('@PLANE2')
    IF (I.EQ.3) STATUS = LIB$SPAWN('@PLANE3')
    IF (I.EQ.4) STATUS = LIB$SPAWN('@PLANE4')
    N=1
    STATUS = PPL$SPAWN(N, '[KHG.SIM.XTMS.F18XX]F18XX.EXE')
    STATUS = PPL$WAIT_AT_BARRIER(BARRIER_INT)
99 CONTINUE
C
STATUS = PPL$CREATE_BARRIER(BARRIER_DATA, 'BARRIER_DATA',
    %REF(COPIES+1))
STATUS = PPL$CREATE_BARRIER(BARRIER_CMD, 'BARRIER_CMD',
    %REF(COPIES+1))
TSTP = 90.0
ISTEP = TSTP * 32
INITIAL = 1
C
C OPERATE LOOP
C
DO 101 I = 0, ISTEP
C
STATUS = PPL$WAIT_AT_BARRIER(BARRIER_DATA)
C
CALL PKILL( RANGE,
    RANGE_RATE,
    LOS,
    AZIMUTH,
    DEVIATION,
    ANGLE_OFF,
    MIS_PK,
    GUN_PK)
C

```

```

CALL TMS_1(desired input, A_COM1, B_COM1, THROT_COM1, SPDBRK_COM1)
    COM_ALPHA(1) = A_COM1
    COM_BANK(1)  = B_COM1
    COM_THRUST(1) = THROT_COM1
    COM_SPDBRK(1) = SPDBRK_COM1
C
CALL TMS_2(desired input, A_COM2, B_COM2, THROT_COM2, SPDBRK_COM2)
    COM_ALPHA(2) = A_COM2
    COM_BANK(2)  = B_COM2
    COM_THRUST(2) = THROT_COM2
    COM_SPDBRK(2) = SPDBRK_COM2
C
CALL TMS_3(desired input, A_COM3, B_COM3, THROT_COM3, SPDBRK_COM3)
    COM_ALPHA(3) = A_COM3
    COM_BANK(3)  = B_COM3
    COM_THRUST(3) = THROT_COM3
    COM_SPDBRK(3) = SPDBRK_COM3
C
CALL TMS_4(desired input, A_COM4, B_COM4, THROT_COM4, SPDBRK_COM4)
    COM_ALPHA(4) = A_COM4
    COM_BANK(4)  = B_COM4
    COM_THRUST(4) = THROT_COM4
    COM_SPDBRK(4) = SPDBRK_COM4
C
STATUS = PPL$WAIT_AT_BARRIER(BARRIER_CMD)
C
101 CONTINUE
END

```

Appendix B

Communication and Synchronization Subroutine

This appendix presents the communication and synchronization subroutine. This subroutine allows the TMS executive program to pass information in and out of the aircraft simulations by means of the shared common block variables. The barriers in this subroutine allow the executive program to maintain synchronization of the simulations.

```
SUBROUTINE TMS
C
C OUTPUT FROM AIRCRAFT SIMULATION
C
1      (ALFDG, BNKCUR, BETDG, CXX, CXY, CXZ, CYX, CYY,
1      CYZ, CZX, CZY, CZZ, PHIDG, THEDG, PSIDG, GAMDG,
1      AZ, MACH, SX, SY, H, XD, YD, HD, EO, E1, E2, E3,
1      PDG, QDG, RDG, PWDG, QWDG, RWDG, DSB, XPCA, XPCS,
1      PCR, T, TT, VT,
C
C INPUT FROM DECISION LOGIC
C
1      ALFCOM ,BNKCOM, CSB, CPR)
C EXTERNAL DEFINITIONS
INTEGER*4 PPL$GET_INDEX
INTEGER*4 PPL$CREATE_BARRIER, PPL$WAIT_AT_BARRIER
INTEGER*4 LIB$STOP, LIB$PUT_OUTPUT
C LOCAL DATA
REAL      MACH
INTEGER*4 STATUS, MY_INDEX
INTEGER*4 ONE_PAGE
      PARAMETER (ONE_PAGE = 512)
C DATA FOR SHARING
BYTE      FRONT_GUARD(ONE_PAGE)
INTEGER    COPIES
REAL      AIRSPEED(4), ALPHA(4), BANKWND(4), BETA(4)
REAL      DIRCOS(4,9), EULER(4,3), GAMMA(4)
REAL      GLOAD(4), MCH(4), POSITION(4,6)
REAL      QUAT(4,4), ROTRATES(4,6), SPDBRAKE(4)
REAL      STKRUD(4,3), TIME(4), THRUST(4)
C
REAL      COM_ALPHA(4), COM_BANK(4), COM_SPDBRK(4), COM_THRUST(4)
BYTE      REAR_GUARD(ONE_PAGE)
C PUT SHARED DATA IN TO COMMON BLOCK
COMMON /SHARED_DATA/ FRONT_GUARD,
1      COPIES,
1      AIRSPEED, ALPHA, BANKWND, BETA,
1      DIRCOS, EULER, GAMMA,
1      GLOAD, MCH, POSITION,
1      QUAT, ROTRATES, SPDBRAKE,
1      STKRUD, TIME, THRUST,
C
1      COM_ALPHA, COM_BANK,
```

```

1      COM_SPDBRK, COM_THRUST,
1      REAR_GUARD
C
  STATUS = PPL$CREATE_BARRIER(BARRIER_DATA, 'BARRIER_DATA',
1    %REF(COPIES+1))
  STATUS = PPL$CREATE_BARRIER(BARRIER_CMD, 'BARRIER_CMD',
1    %REF(COPIES+1))
  STATUS = PPL$GET_INDEX(MY_INDEX)
C
C***** PASS DATA TO TMS EXECUTIVE*****
C
  AIRSPEED(MY_INDEX)      = VT
  ALPHA(MY_INDEX)         = ALFDG
  BANKWND(MY_INDEX)       = BNKCUR
  BETA(MY_INDEX)          = BETDG
  DIRCOS(MY_INDEX,1)      = CXX
  DIRCOS(MY_INDEX,2)      = CXY
  DIRCOS(MY_INDEX,3)      = CXZ
  DIRCOS(MY_INDEX,4)      = CYX
  DIRCOS(MY_INDEX,5)      = CYY
  DIRCOS(MY_INDEX,6)      = CYZ
  DIRCOS(MY_INDEX,7)      = CZX
  DIRCOS(MY_INDEX,8)      = CZY
  DIRCOS(MY_INDEX,9)      = CZZ
  EULER(MY_INDEX,1)       = PHIDG
  EULER(MY_INDEX,2)       = THEDG
  EULER(MY_INDEX,3)       = PSIDG
  GAMMA(MY_INDEX)         = GAMDG
  GLOAD(MY_INDEX)         = AZ
  MCH(MY_INDEX)           = MACH
  POSITION(MY_INDEX,1)     = SX
  POSITION(MY_INDEX,2)     = SY
  POSITION(MY_INDEX,3)     = -1. * H
  POSITION(MY_INDEX,4)     = XD
  POSITION(MY_INDEX,5)     = YD
  POSITION(MY_INDEX,6)     = -1. * HD
  QUAT(MY_INDEX,1)        = E0
  QUAT(MY_INDEX,2)        = E1
  QUAT(MY_INDEX,3)        = E2
  QUAT(MY_INDEX,4)        = E3
  ROTRATES(MY_INDEX,1)    = PDG
  ROTRATES(MY_INDEX,2)    = QDG
  ROTRATES(MY_INDEX,3)    = RDG
  ROTRATES(MY_INDEX,4)    = PWDG
  ROTRATES(MY_INDEX,5)    = QWDG
  ROTRATES(MY_INDEX,6)    = RWDG
  SPDBRAKE(MY_INDEX)      = DSB / 60.0
  STKRUD(MY_INDEX,1)      = PCA
  STKRUD(MY_INDEX,2)      = PCS
  STKRUD(MY_INDEX,3)      = PCR
  TIME(MY_INDEX)          = T

```

```

        THRUST(MY_INDEX)      = TT
C
    STATUS = PPL$WAIT_AT_BARRIER(BARRIER_DATA)
C
    STATUS = PPL$WAIT_AT_BARRIER(BARRIER_CMD)
C
C*****ACCEPT COMMANDS FROM EXECUTIVE*****
C
    ALFCOM      = COM_ALPHA(MY_INDEX)
    BNKCOM      = COM_BANK(MY_INDEX)
    CSB         = COM_SPDBRK(MY_INDEX)
    CPR         = COM_THRUST(MY_INDEX)
C
    RETURN
C
    END

```

Appendix C

Example Trim File

This appendix presents an input or trim file for defining the initial conditions for a simulated aircraft.

PHYSICAL CONSTANTS :

```
31665.0  WT    (LBS)  -WEIGHT
22337.0  IXX   (SLUG*FT**2) -INERTIA ABOUT X AXIS
120293.0 IYY   (SLUG*FT**2) -INERTIA ABOUT Y AXIS
138945.0 IZZ   (SLUG*FT**2) -INERTIA ABOUT Z AXIS
-2430.0  IXZ   (SLUG*FT**2) -XZ PLANE INERTIA PRODUCT
  457.3  FSCG  (IN)    -FUSELAGE STATION CG
    0.0  BLCG  (IN)    -BUTTOCK LINE CG
  101.6  WLCG  (IN)    -WATER LINE CG
  687.5  XNRF  (IN)    -X THRUST CENTERLINE
   18.9  YNRF  (IN)    -Y THRUST CENTERLINE
  100.0  ZNRF  (IN)    -Z THRUST CENTERLINE
```

FLIGHT CONDITIONS AND MODELING OPTIONS:

```
F LTHVEC >> TRUE= THRUST VECTOR ON
T LFCS >> TRUE= FLT CONTROL SYTEM ON
T LTHDMS >> TRUE= DMS PLA SCHEDULE
T LRTE >> TRUE= R/T EQV AERO
  0.90000  MACHTR (N.D.)
10000.00  HIC    (FT)
  5000.00  X IC   (FT)
    0.00  Y IC   (FT)
    0.00  MUDGTR (DEG)
  1.00000  GLOAD  (G)
```

TCASE:

1

TRIM DRIVER VALUES:

NXTR -NUMBER OF DRIVER VARIABLES

4

ELEMENT	LIMITS	VARIABLE NAME, UNITS
3	0.000 1.000	ALFTR (RADIAN)
7	-1.000 1.000	THETR (RADIAN)
14	-2.500 5.000	PCSTR (INCHES)
16	31.000 130.000	DPSYTR (% POWER)

TRIM DRIVEN VALUES:

NYTR -NUMBER OF DRIVEN VARIABLES

4

ELEMENT	VARIABLE NAME, UNITS
---------	----------------------

1	UD (FT/SEC2)
3	WD (FT/SEC2)
5	QD (RAD/SEC2)
7	GAMZR (RADIAN)

NFSY -OLD VAR RETAINED FOR FILE COMPATIBILITY

0

NFAS -OLD VAR RETAINED FOR FILE COMPATIBILITY

0

INITIAL CONDITIONS:

0.90000000	MACHTR	(N.D.)
0.00000000E+00	BETTR	(RADIANS)
0.45296673E-01	ALFTR	(RADIANS)
0.00000000E+00	PIC	(RAD/SEC)
0.00000000E+00	QIC	(RAD/SEC)
0.00000000E+00	RIC	(RAD/SEC)
0.45298599E-01	THETR	(RADIANS)
0.00000000E+00	PHITR	(RADIANS)
3.14159265E+00	PSITR	(RADIANS)
0.00000000E+00	GAMTR	(RADIANS)
0.00000000E+00	DTVL	(DEGREES)
0.00000000E+00	DTVR	(DEGREES)
0.50865169E-05	PCATR	(INCHES)
0.24973108E-01	PCSTR	(INCHES)
0.61914313E-03	PCRTR	(LBS)
77.252785	DPSYTR	(% POWER)
0.00000000E+00	DPASTR	(% POWER)
-0.18907314E-01	DSSYTR	(DEGREES)
-0.12405217E-05	DSASTR	(DEGREES)
0.00000000E+00	DASYTR	(DEGREES)
-0.31018224E-05	UD	(FT/SEC2)
0.00000000E+00	DRSYTR	(DEGREES)
-0.15484100E-04	DRASTR	(DEGREES)
3.4465680	DNSYTR	(DEGREES)
0.00000000E+00	DNASTR	(DEGREES)
3.6334312	DFSYTR	(DEGREES)
-0.95367432E-06	DFASTR	(DEGREES)
0.00000000E+00	CSB	(DEGREES)

\$END OF DATA READING SECTION

CASE SELECTIONS:

TCASE 1 STRAIGHT & LEVEL STEADY STATE
 2 COORDINATED TURN STEADY STATE
 3 PULL-UP STEADY STATE

TRIM VALUE SELECTIONS:

TRIM DRIVER ARRAY	TRIM OUTPUT ARRAY
1 MACHTR (N.D.)	1 UD (FT/SEC2)
2 BETTR (RADIANS)	2 VD (FT/SEC2)
3 ALFTR (RADIANS)	3 WD (FT/SEC2)
4 PIC (RAD/SEC)	4 PD (RAD/SEC2)
5 QIC (RAD/SEC)	5 QD (RAD/SEC2)
6 RIC (RAD/SEC)	6 RD (RAD/SEC2)
7 THETR (RADIANS)	7 GAMZR (RADIANS)
8 PHITR (RADIANS)	8 PHIZR (RADIANS)
9 PSITR (RADIANS)	9 THE (RADIANS)
10 GAMTR (RADIANS)	10 LAMDA (RADIANS)
11 DTVL (DEGREES)	11 FYTOT (G S)
12 DTVR (DEGREES)	
13 PCATR (INCHES)	
14 PCSTR (INCHES)	
15 PCRTR (LBS)	
16 DPSYTR (% POWER)	

17	DPASTR	(% POWER)
18	DSSYTR	(DEGREES)
19	DSASTR	(DEGREES)
20	DASYTR	(DEGREES)
21	DAASTR	(DEGREES)
22	DRSYTR	(DEGREES)
23	DRASTR	(DEGREES)
24	DNSYTR	(DEGREES)
25	DNASTR	(DEGREES)
26	DFSYTR	(DEGREES)
27	DFASTR	(DEGREES)
28	CSB	(DEGREES)

Table I. Summary of Weight, Center of Gravity, and Inertia

Weight, lb	Center-of-gravity locations		Moments and product of inertia, slugs/ft ²			
	Fuselage station, in.	Water line, in.	I_{XX}	I_{YY}	I_{ZZ}	I_{XZ}
TV aircraft						
33 310	455.0	102.8	23 000	151 293	169 945	-2971
Baseline aircraft						
31 665	457.3	101.6	22 337	120 293	138 945	-2430

Table II. Dimensional Data^a

Total airplane:	
Net wetted area (minus engine nozzles), ft ²	2028
Overall length, ft	56.0
Overall height, ft	15.3
Wing:	
Area, S_{REF} , ft ²	400
Wetted area (including launchers and aileron actuator fairings), ft	562
Span, b_{REF} , ft	37.42
Aspect ratio	3.5
\bar{c}_{REF} , ft	11.52
Leading-edge sweep, deg	26.7
$\bar{c}/4$ sweep, deg	20
Taper ratio	0.35
Dihedral, deg	-3
Leading-edge flaps:	
Deflection (positive leading edge down), deg—	
Maneuvering	0, 34
Takeoff and landing	12, 34
Differential	±3
Actuator 18 deg/sec rate limit	1/(s/20 + 1)
Trailing-edge flaps:	
Deflections (positive trailing edge down), deg—	
Takeoff	+17, +30
Landing	+17, +45
Actuator 18 deg/sec rate limit	1/(s/20 + 1)

^aFrom reference 13.

Table II. Concluded

Ailerons:

Deflections (positive trailing edge down), deg—

Takeoff and landing	-25, +45
Maneuvering	-25, +25
Actuator 100 deg/sec rate limit	$1/(s/48 + 1)$

Horizontal tails (HT):

Exposed area, ft ²	88.1
Aspect ratio	2.4
$\bar{c}/4$ sweep, deg	42.8
Span, b_{REF} , ft	14.67
\bar{c}_{HT} , ft	6.28
Deflections (positive trailing edge down), deg—	
Symmetric	-24, +8
Maximum	-24, +10.5
Actuator 40 deg/sec rate limit	$1/(s/30 + 1)$

Vertical tails (VT):

Area, ft ²	52.0 each
Wetted area, ft ²	104.0 each
$\bar{c}/4$ sweep, deg	35.0
Cant (tip out), deg	20
\bar{c}_{VT} , ft	6.99
Tail length ($\bar{c}/4$ to $\bar{c}_{VT}/4$), ft	10.18

Rudders:

Deflection, deg	± 30
Actuator 61 deg/sec rate limit	$1/(s/40 + 1)$

Speedbrake:

Planform area, ft ²	13.9
Span, ft	2.5
Chord, ft	5.57
Maximum deflection, deg	60

Table III. Time Required by TA To Perform α Captures[All runs started at $h = 25\,000$ ft and had $\pm 2^\circ$ capture criteria]

Initial α , deg	Final α , deg	Initial M	Average time by pilot, sec	Minimum time by pilot, sec	Time by TA, sec	Maximum overshoot, deg
Baseline aircraft						
4.4	30.0	0.60	5.12	4.35	1.91	1.9
4.4	40.0	.60	2.88	2.30	^a 2.28	2.4
23.5	30.0	.30	4.93	3.78	1.00	1.4
23.5	40.0	.30	6.56	5.95	1.81	1.6
10.0	0.0	.40	2.50	1.99	1.34	1.0
20.0	0.0	.32	5.86	5.25	1.88	2.0
30.0	0.0	.27	7.06	5.68	2.38	2.0
TV aircraft						
4.4	30.0	0.60	4.70	3.84	1.09	1.7
4.4	40.0	.60	4.45	3.46	2.97	2.6
4.4	50.0	.60	4.76	5.31	2.41	.2
23.5	30.0	.30	2.11	1.09	.81	1.2
23.5	40.0	.30	2.69	1.41	1.38	1.2
23.5	50.0	.30	3.39	1.79	1.78	1.6
10.0	0.0	.40	2.18	2.18	1.12	.4
20.0	0.0	.32	2.11	1.66	1.60	.7
30.0	0.0	.27	4.60	4.54	1.89	.6

^aCapture criteria relaxed to $\pm 2.4^\circ$.Table IV. Time Required by TA To Perform $90^\circ \mu$ Captures[All runs started at $h = 25\,000$ ft, Initial $\mu = 0^\circ$, and Final $\mu = 90^\circ$]

Initial α , deg	Capture criteria, deg	Average time by pilot, sec	Minimum time by pilot, sec	Time by TA, sec	Maximum overshoot, deg
Baseline aircraft					
10	± 5	4.10	3.07	1.43	3.8
20	± 8	8.90	6.70	4.90	6.0
TV aircraft					
10	± 5	2.15	1.47	1.53	2.8
20	± 5	5.00	4.40	2.22	2.7
30	± 5	5.17	2.75	2.50	3.9

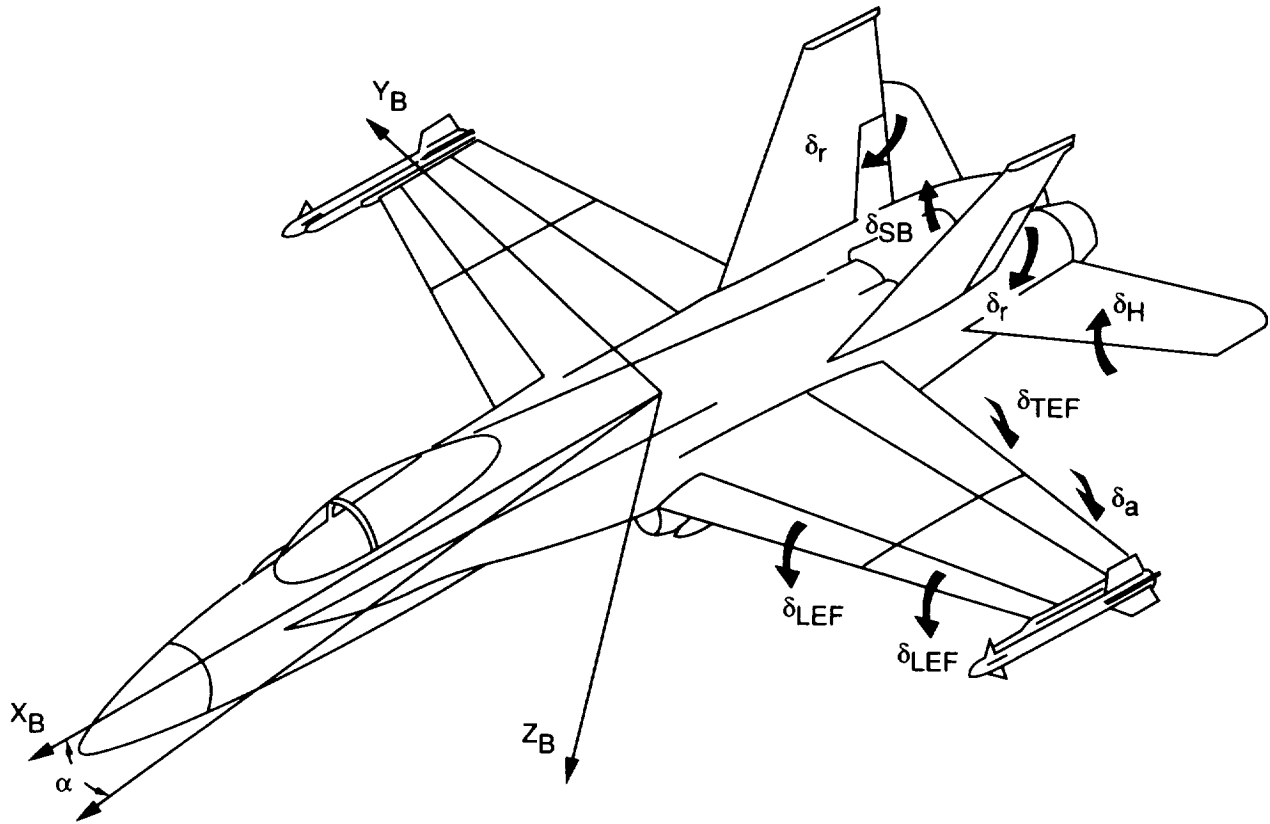


Figure 1. Configuration of aerodynamic surfaces, definitions of axes, and sign convention (ref. 14).

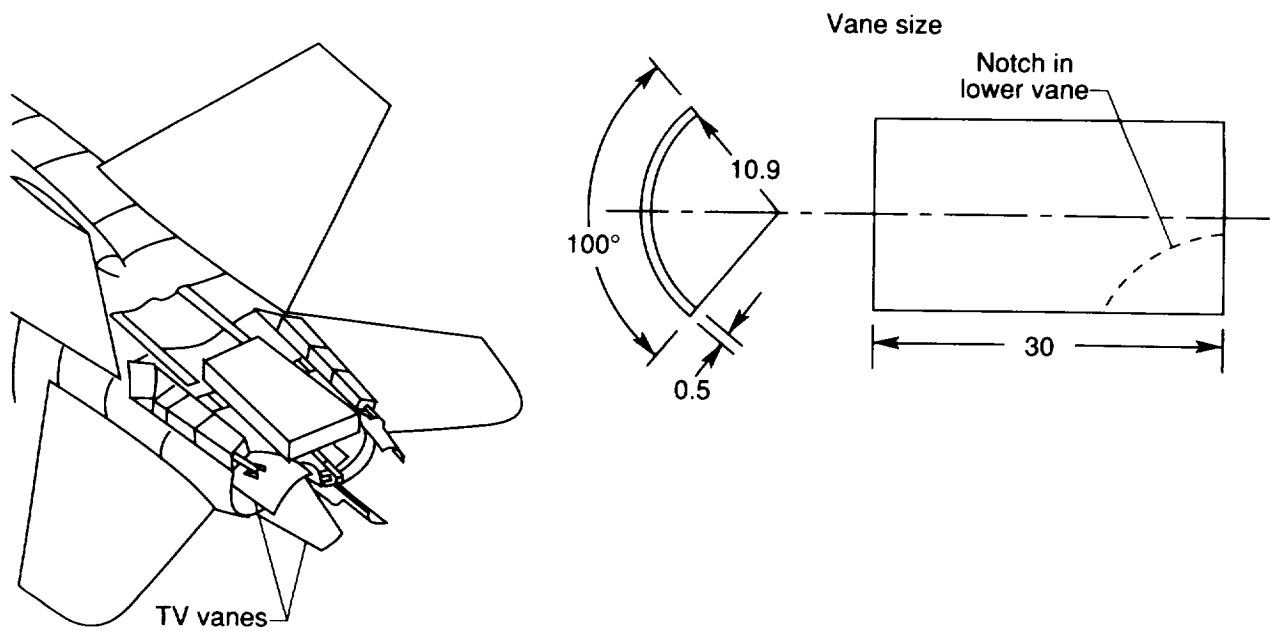


Figure 2. TV system. Vane cant angle 48° ; maximum vane deflection $\pm 30^\circ$; maximum deflection rate 80 deg/sec. (All linear dimensions are in inches.)

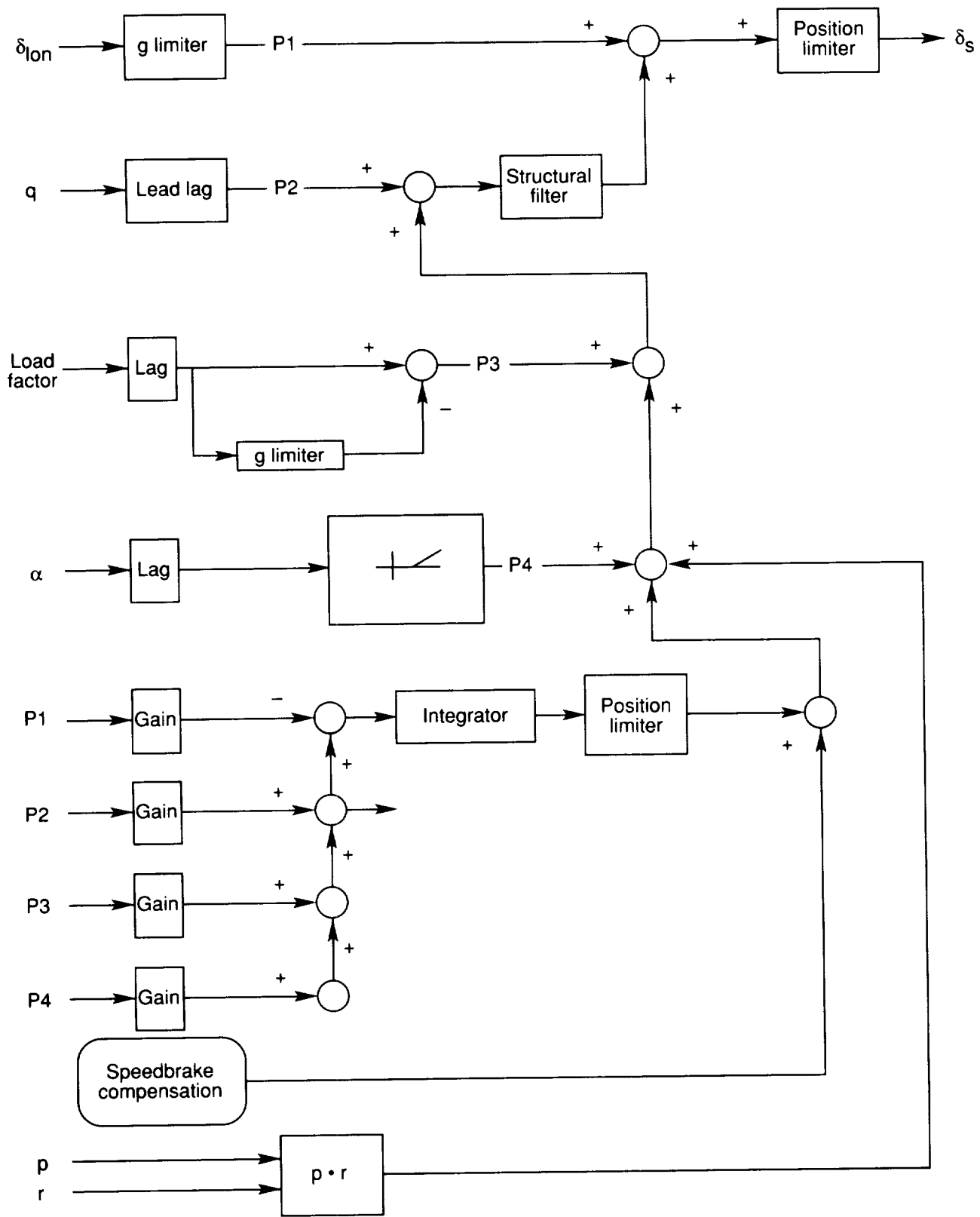


Figure 3. P = Signal. Longitudinal control augmentation system (ref. 20).

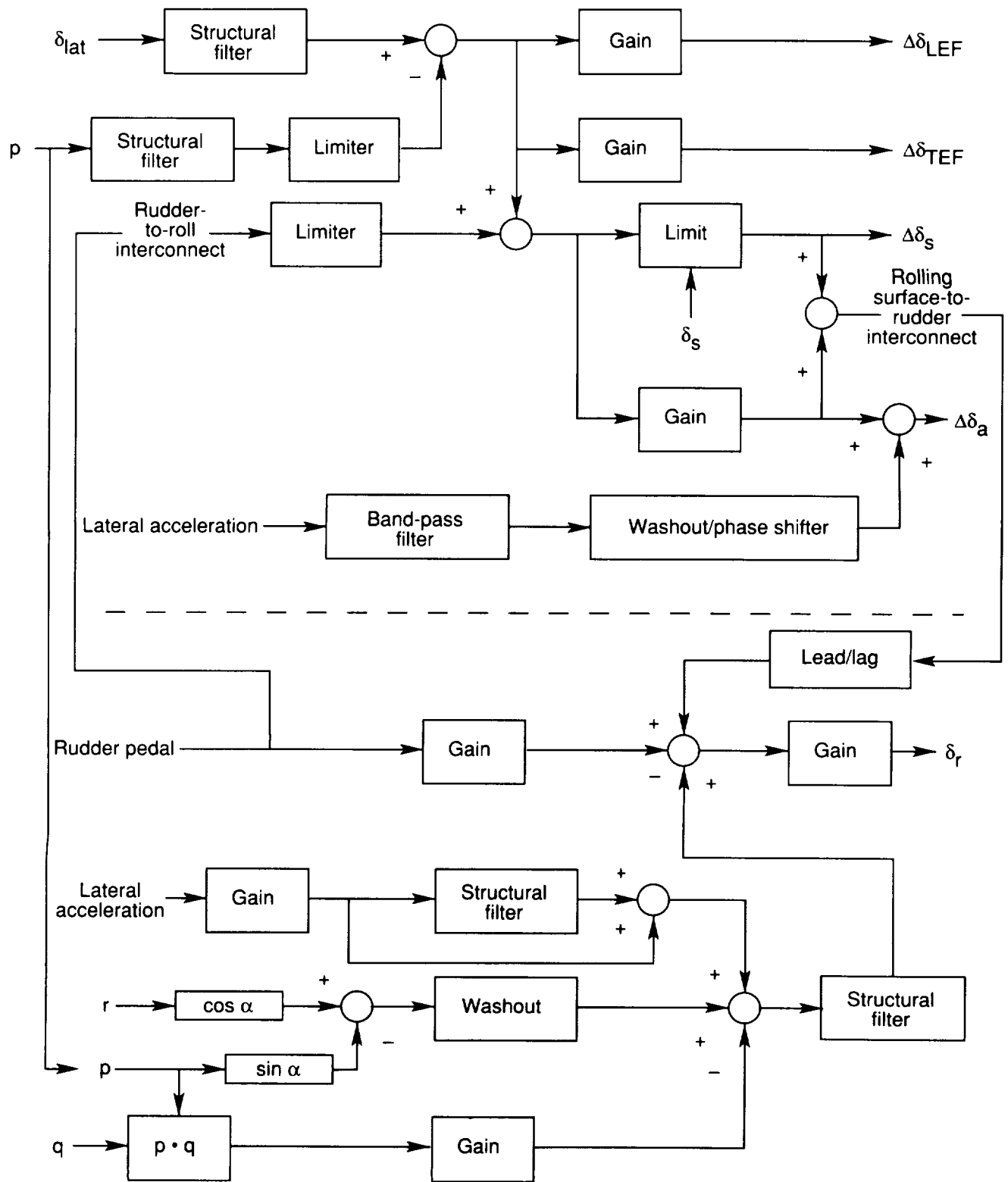


Figure 4. Lateral and directional control augmentation system (ref. 20).

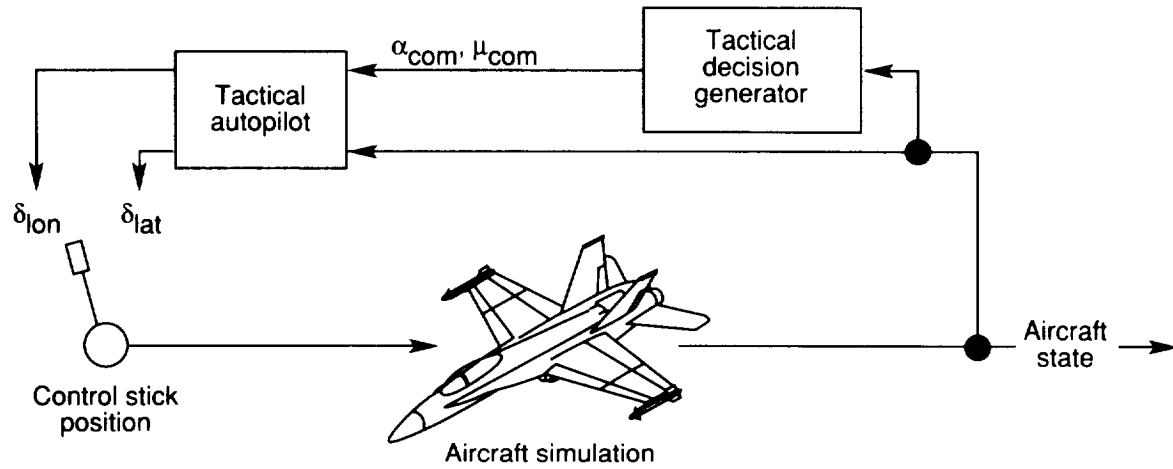
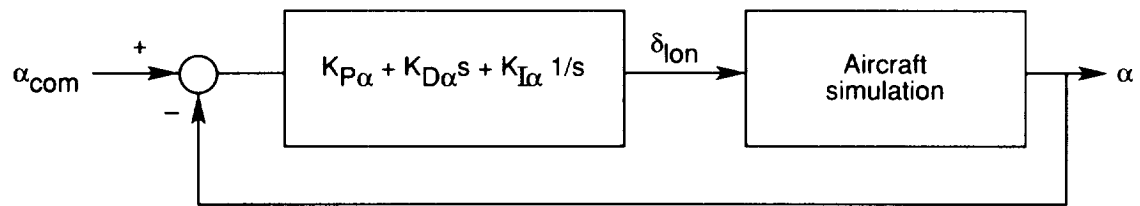
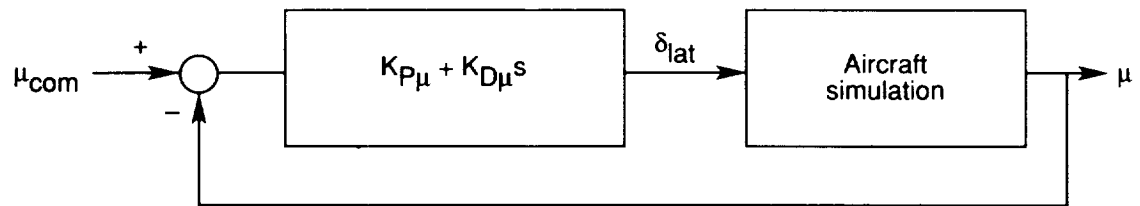


Figure 5. TDG TA aircraft system.

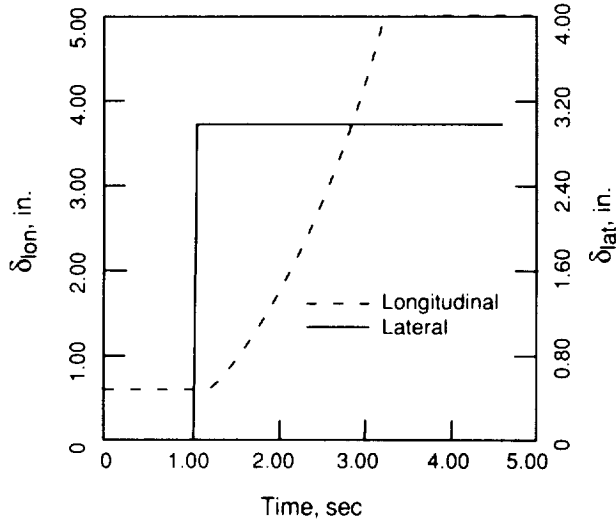


(a) Longitudinal command system.

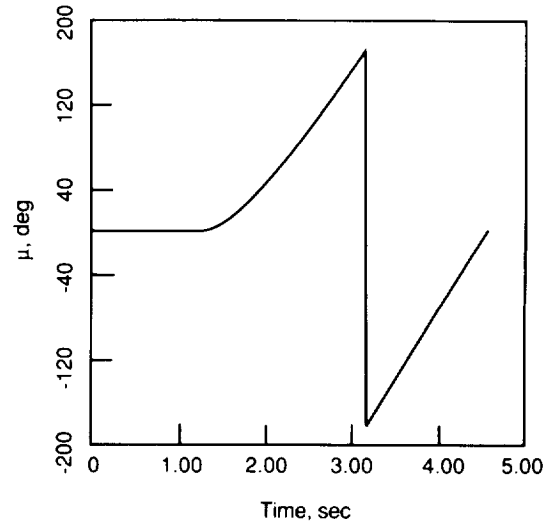


(b) Lateral command system.

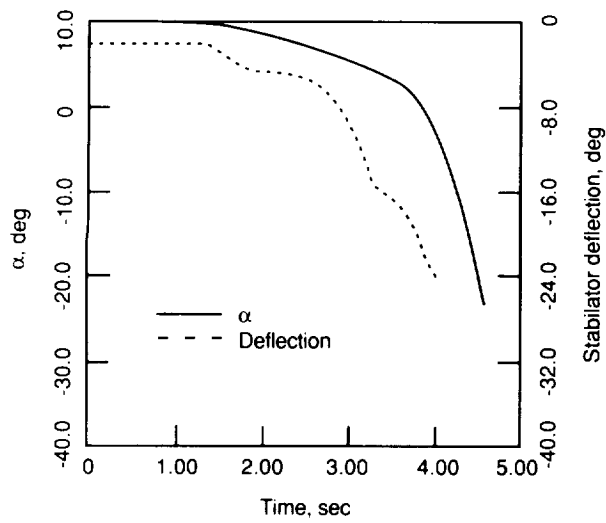
Figure 6. Block diagram of TA system.



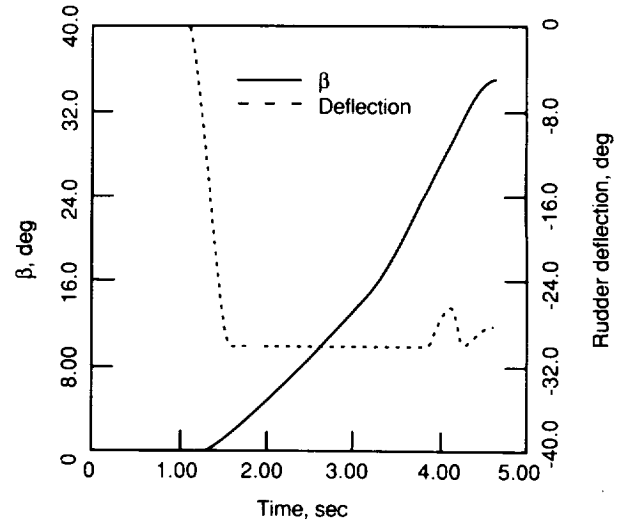
(a) Control inputs.



(b) Wind-axis bank-angle response.

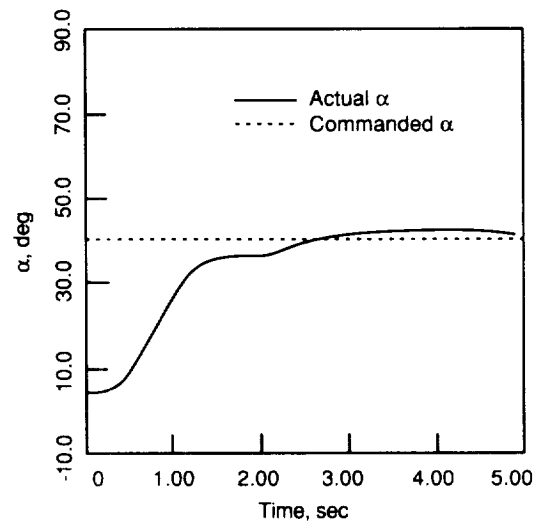
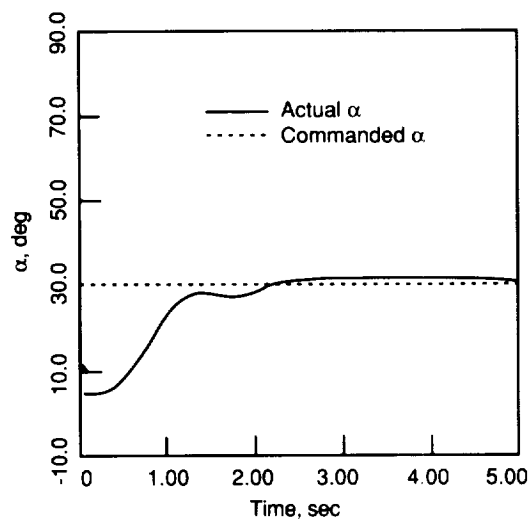


(c) Angle-of-attack and stabilator responses up to $\alpha = -10^\circ$.

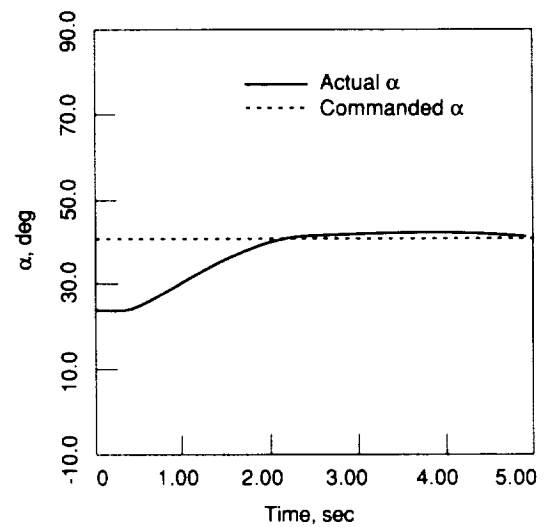
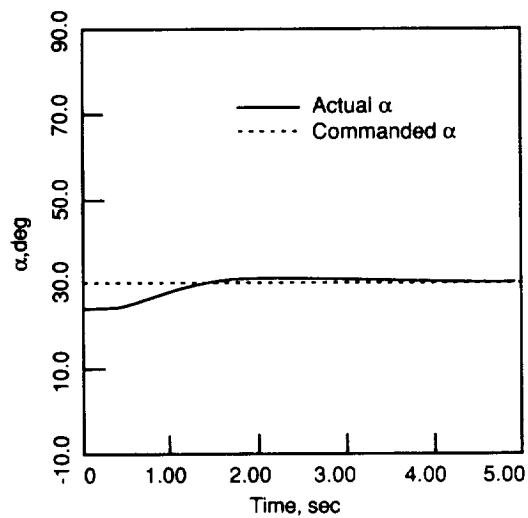


(d) Angle-of-sideslip and rudder responses up to $\beta = 20^\circ$.

Figure 7. Departure of baseline aircraft with full lateral stick input. Initial $M = 0.30$; $h = 10\,000$ ft.

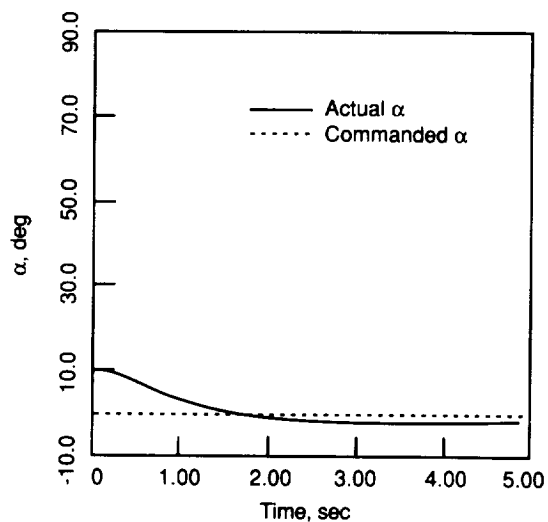


(a) Initial $M = 0.60$, baseline aircraft.

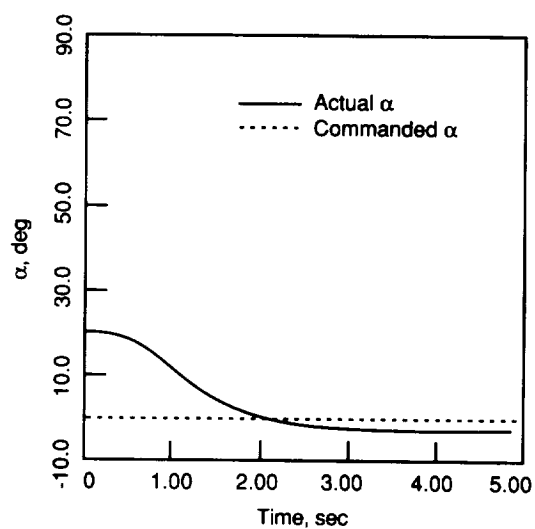


(b) Initial $M = 0.30$, baseline aircraft.

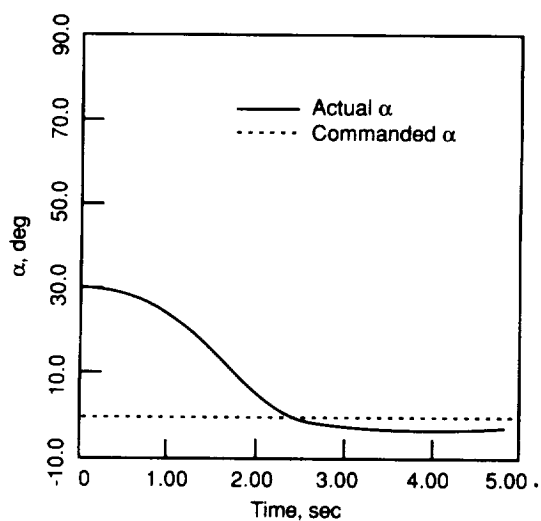
Figure 8. Performance of TA for longitudinal captures.



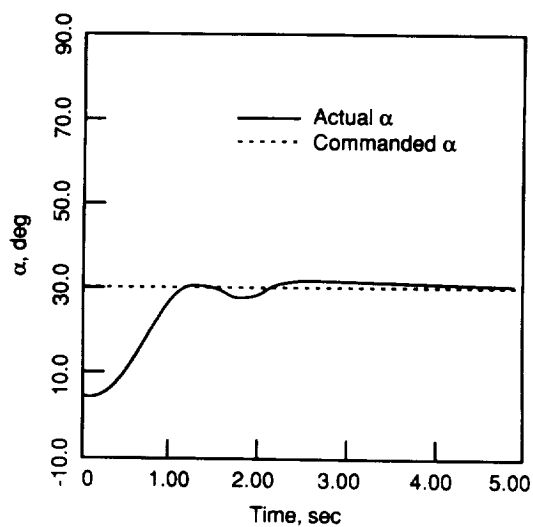
(c) Initial $M = 0.40$, baseline aircraft.



(d) Initial $M = 0.32$, baseline aircraft.

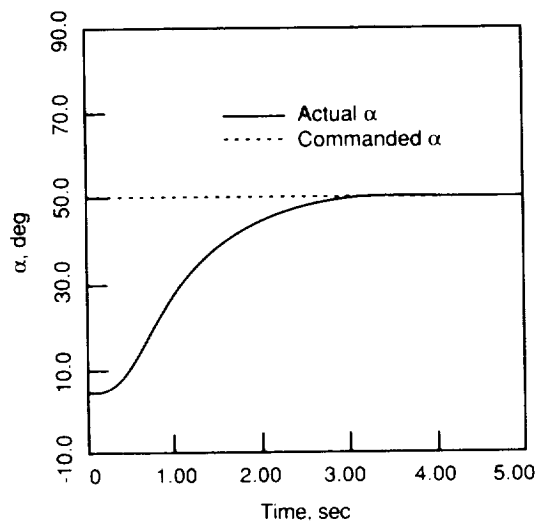
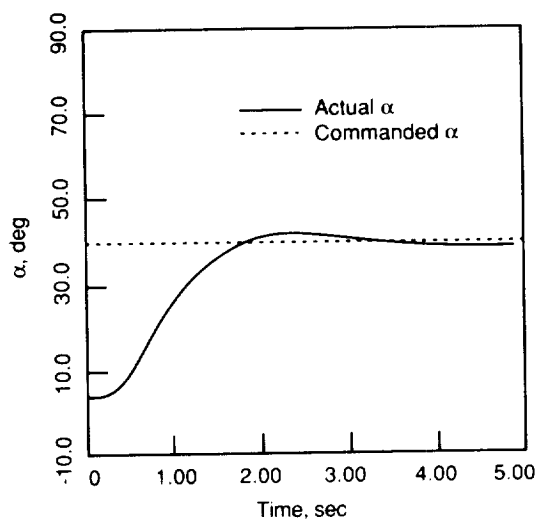


(e) Initial $M = 0.27$, baseline aircraft.

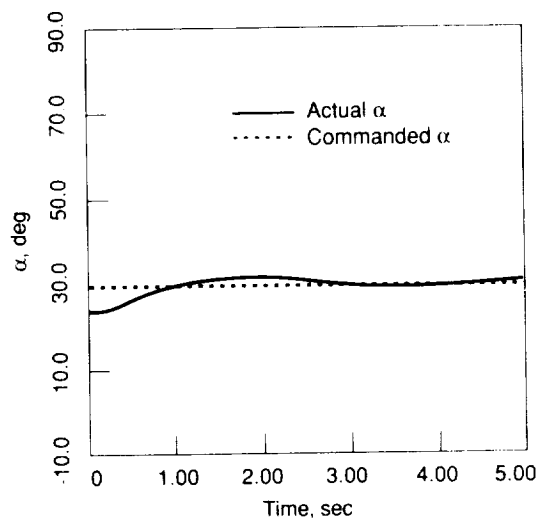


(f) Initial $M = 0.60$, TV aircraft.

Figure 8. Continued.

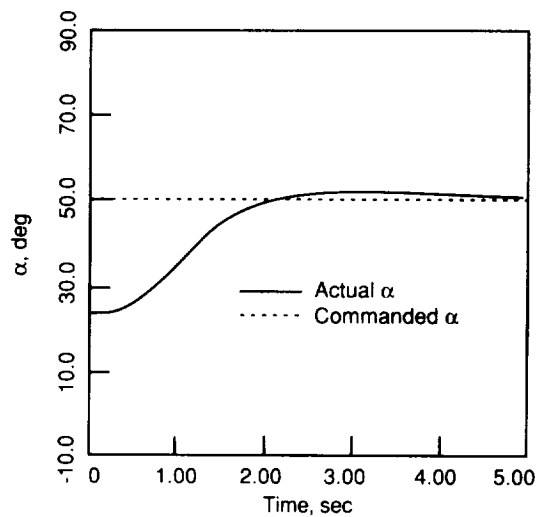


(g) Initial $M = 0.60$, TV aircraft.

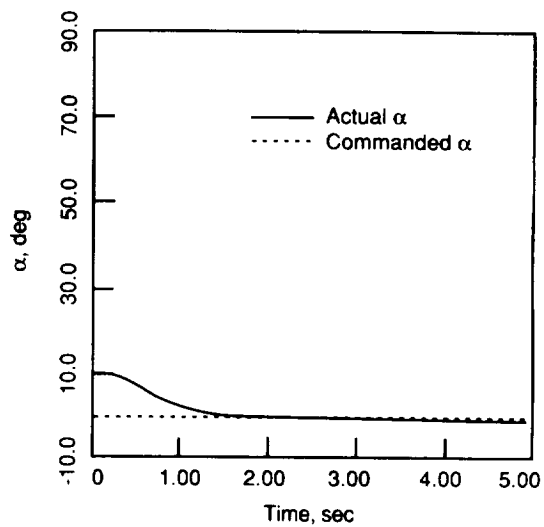


(h) Initial $M = 0.30$, TV aircraft.

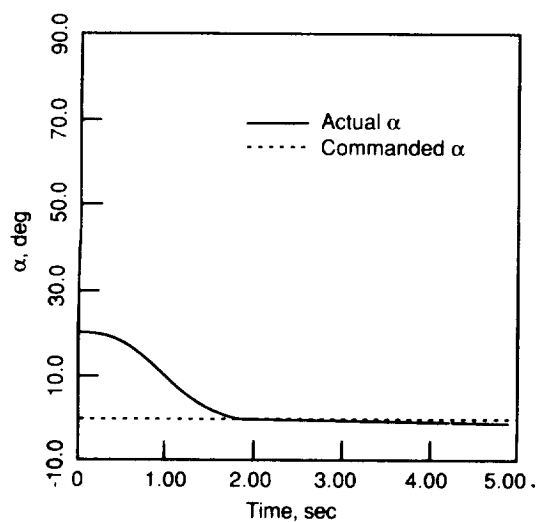
Figure 8. Continued.



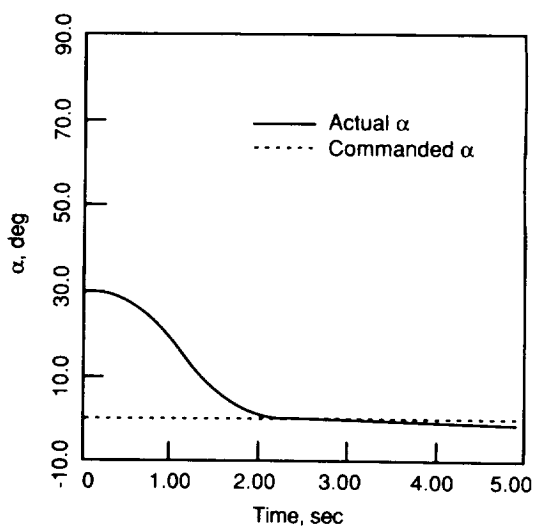
(i) Initial $M = 0.30$, TV aircraft.



(j) Initial $M = 0.40$, TV aircraft.

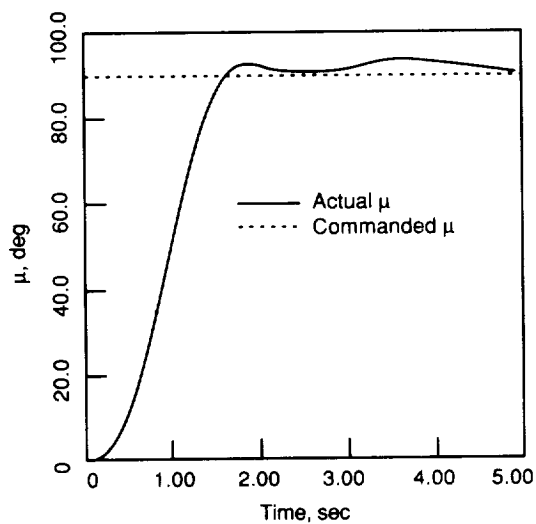


(k) Initial $M = 0.32$, TV aircraft.

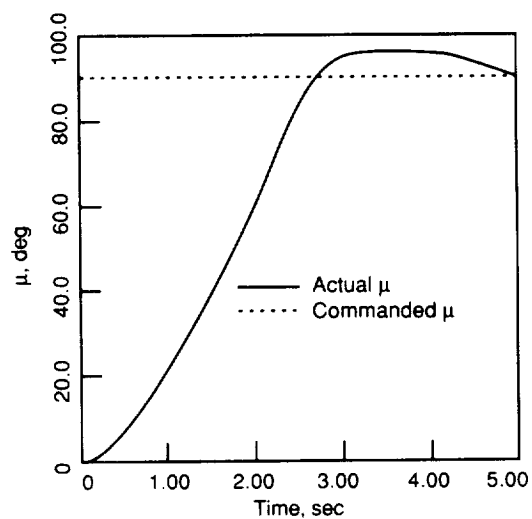


(l) Initial $M = 0.27$, TV aircraft.

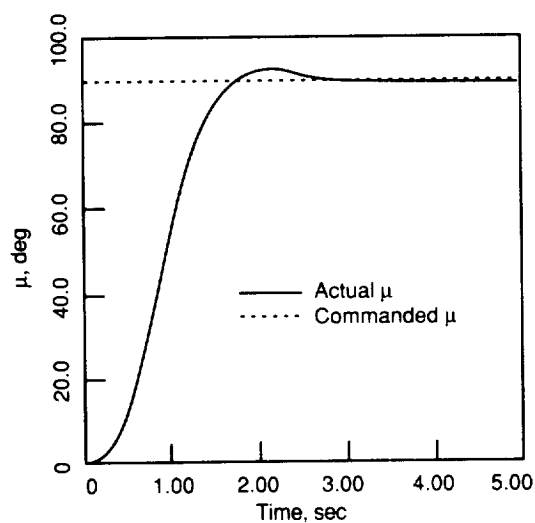
Figure 8. Concluded.



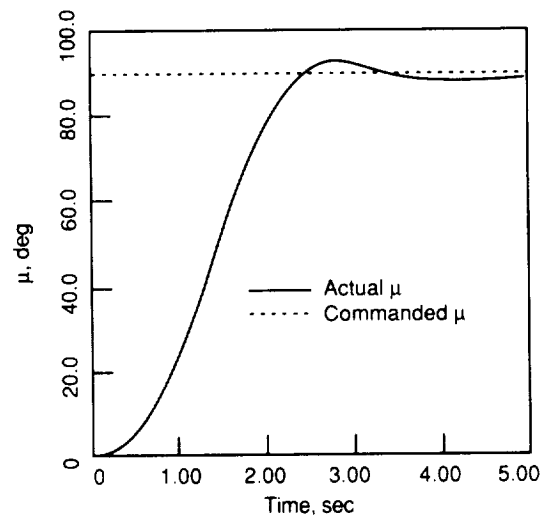
(a) $\alpha = 10^\circ$, baseline aircraft.



(b) $\alpha = 20^\circ$, baseline aircraft.

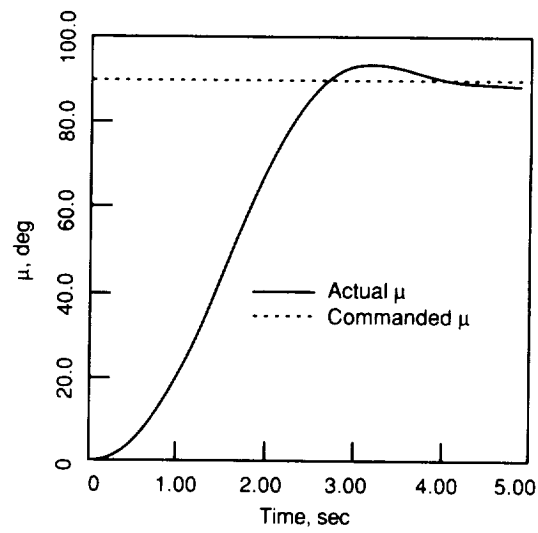


(c) $\alpha = 10^\circ$, TV aircraft.



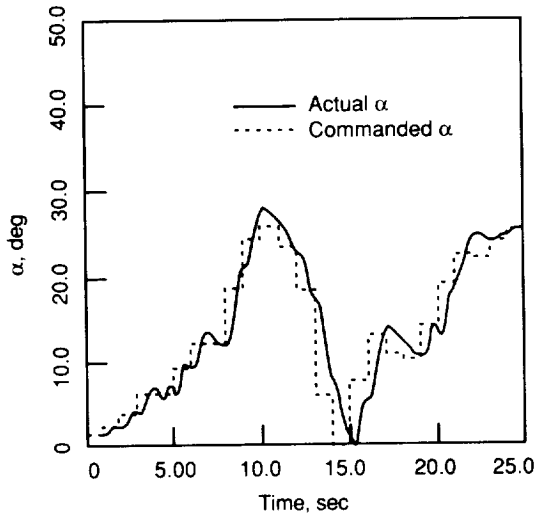
(d) $\alpha = 20^\circ$, TV aircraft.

Figure 9. Performance of TA for lateral captures.

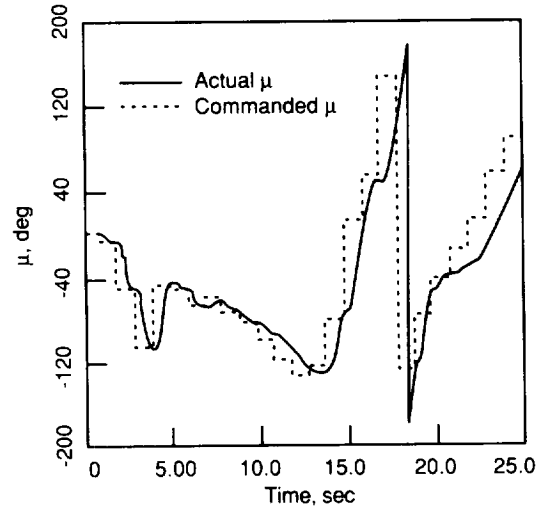


(e) $\alpha = 30^\circ$, TV aircraft.

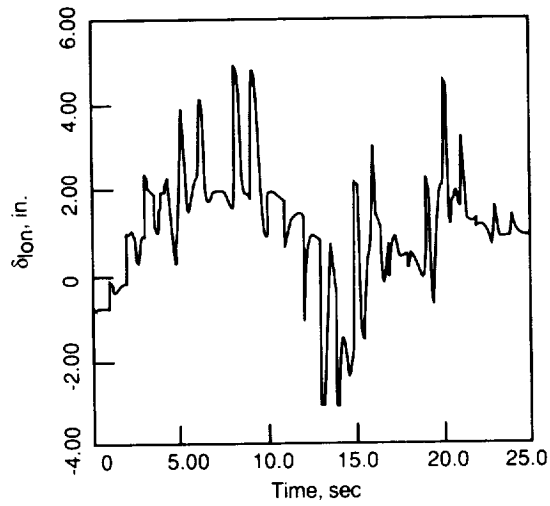
Figure 9. Concluded.



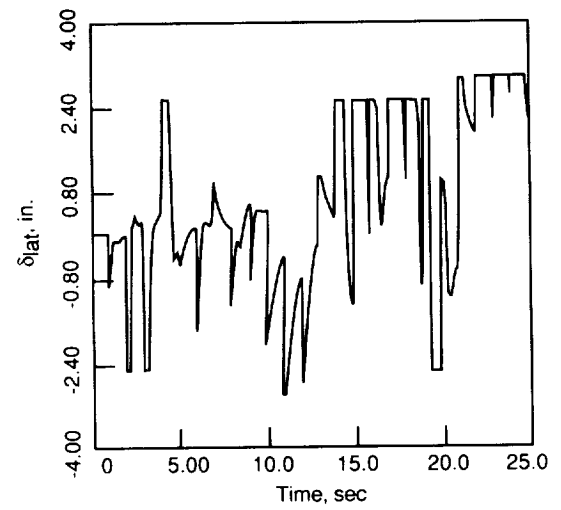
(a) α response.



(b) Wind-axis bank-angle response.

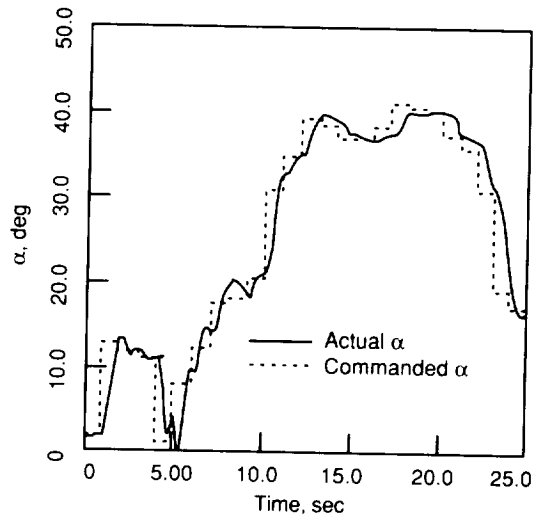


(c) Longitudinal stick deflection.

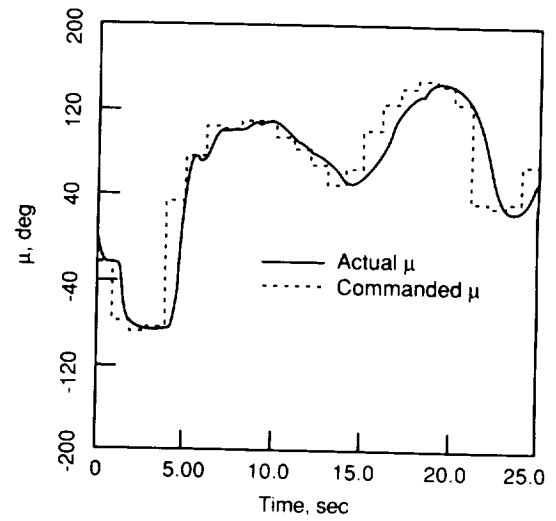


(d) Lateral stick deflection.

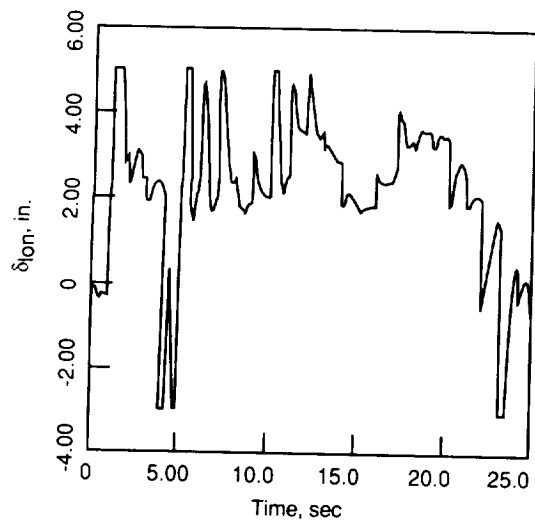
Figure 10. Response of TA to command sequence for baseline aircraft.



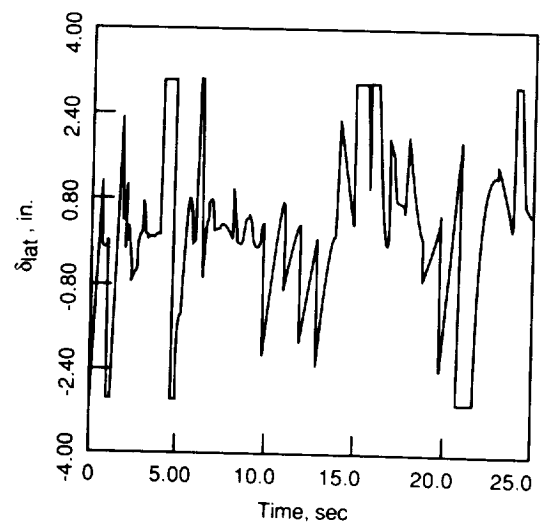
(a) α response.



(b) Wind-axis bank-angle response.



(c) Longitudinal stick deflection.



(d) Lateral stick deflection.

Figure 11. Response of TA to command sequence for TV aircraft.

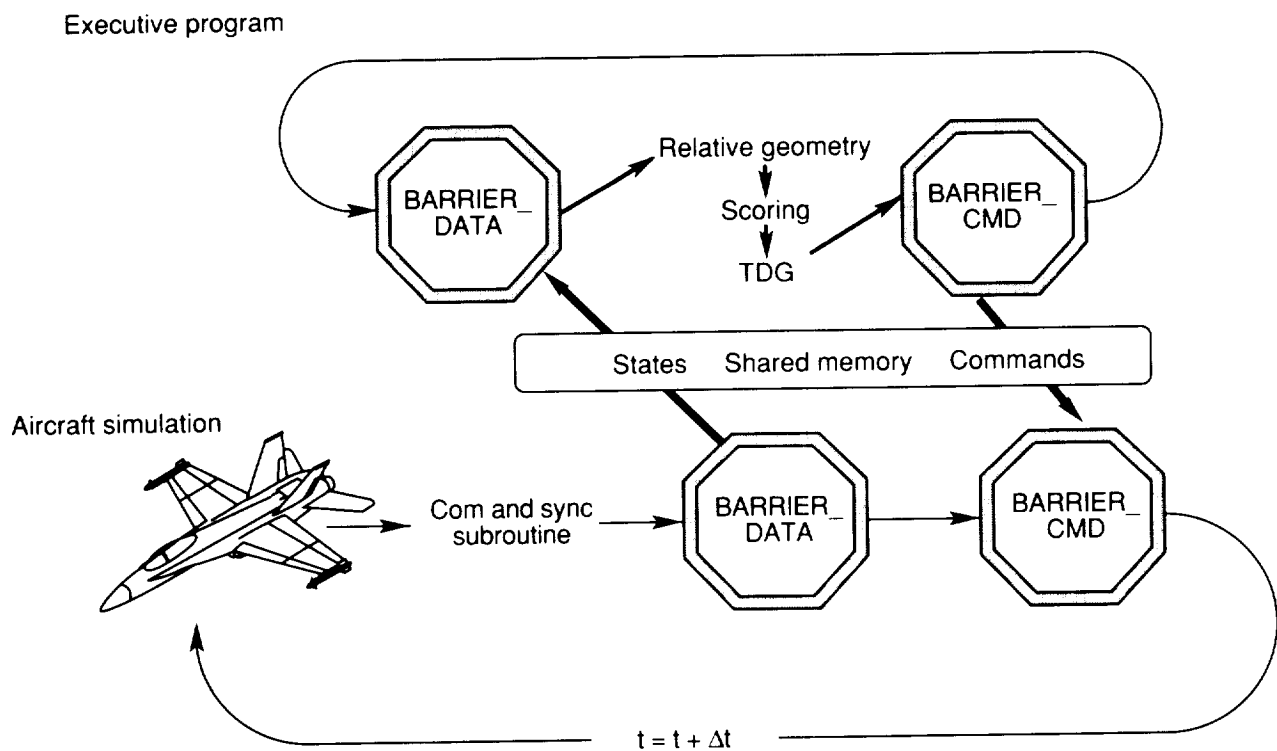


Figure 12. Operational flow between TMS executive program and one aircraft simulation. Additional aircraft are simulated by concurrent execution of multiple copies of aircraft simulation model.

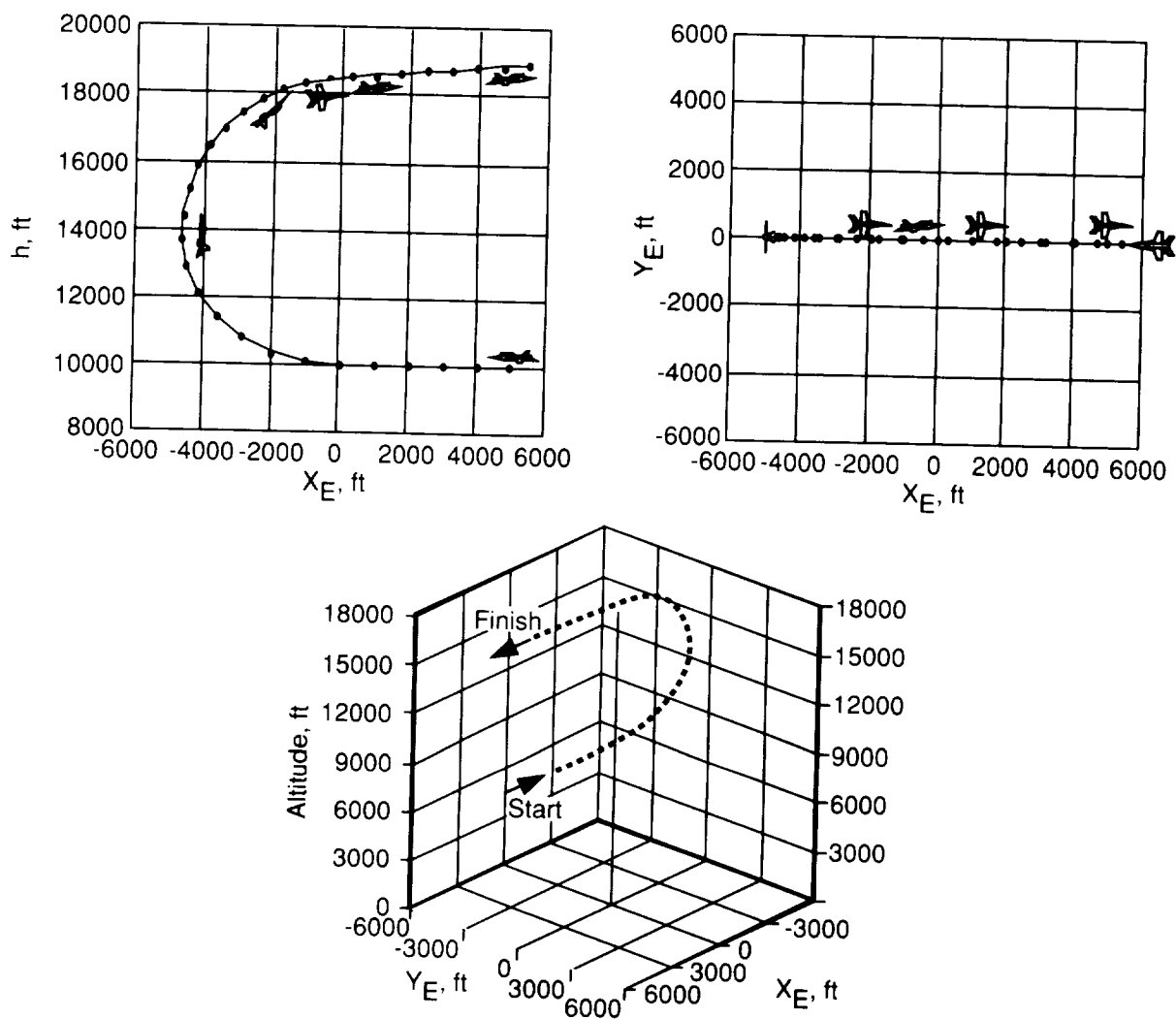


Figure 13. Vertical reversal maneuver; position plotted at 1-sec intervals.

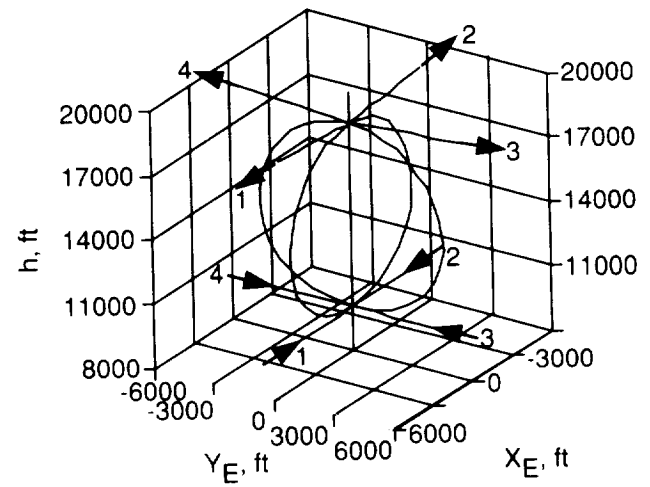
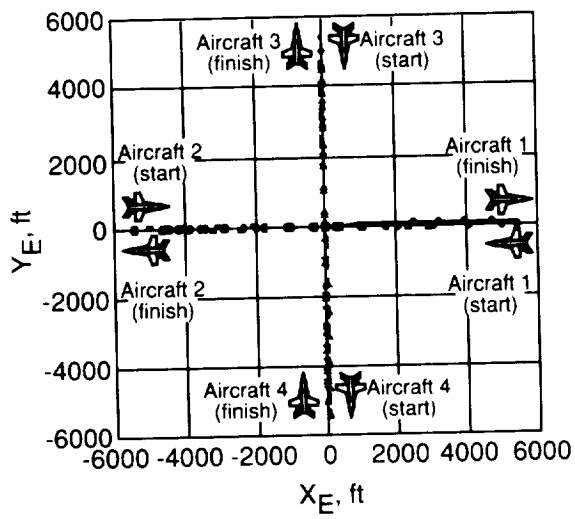
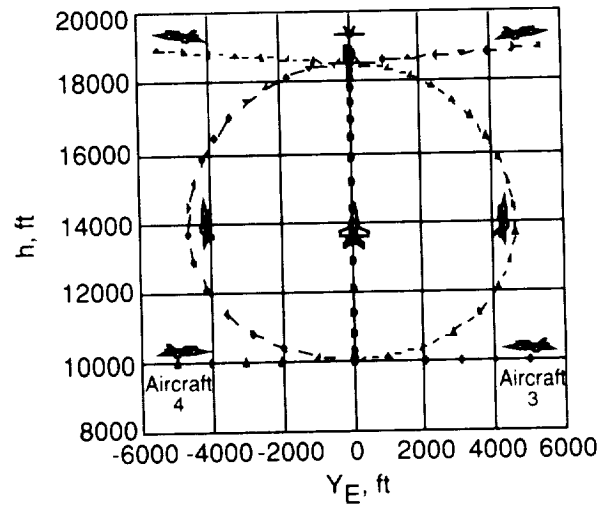
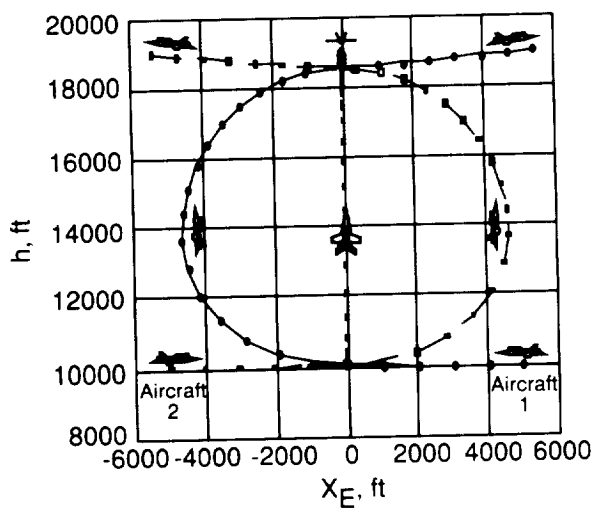


Figure 14. Simulation of multiple aircraft; positions plotted at 1-sec intervals.

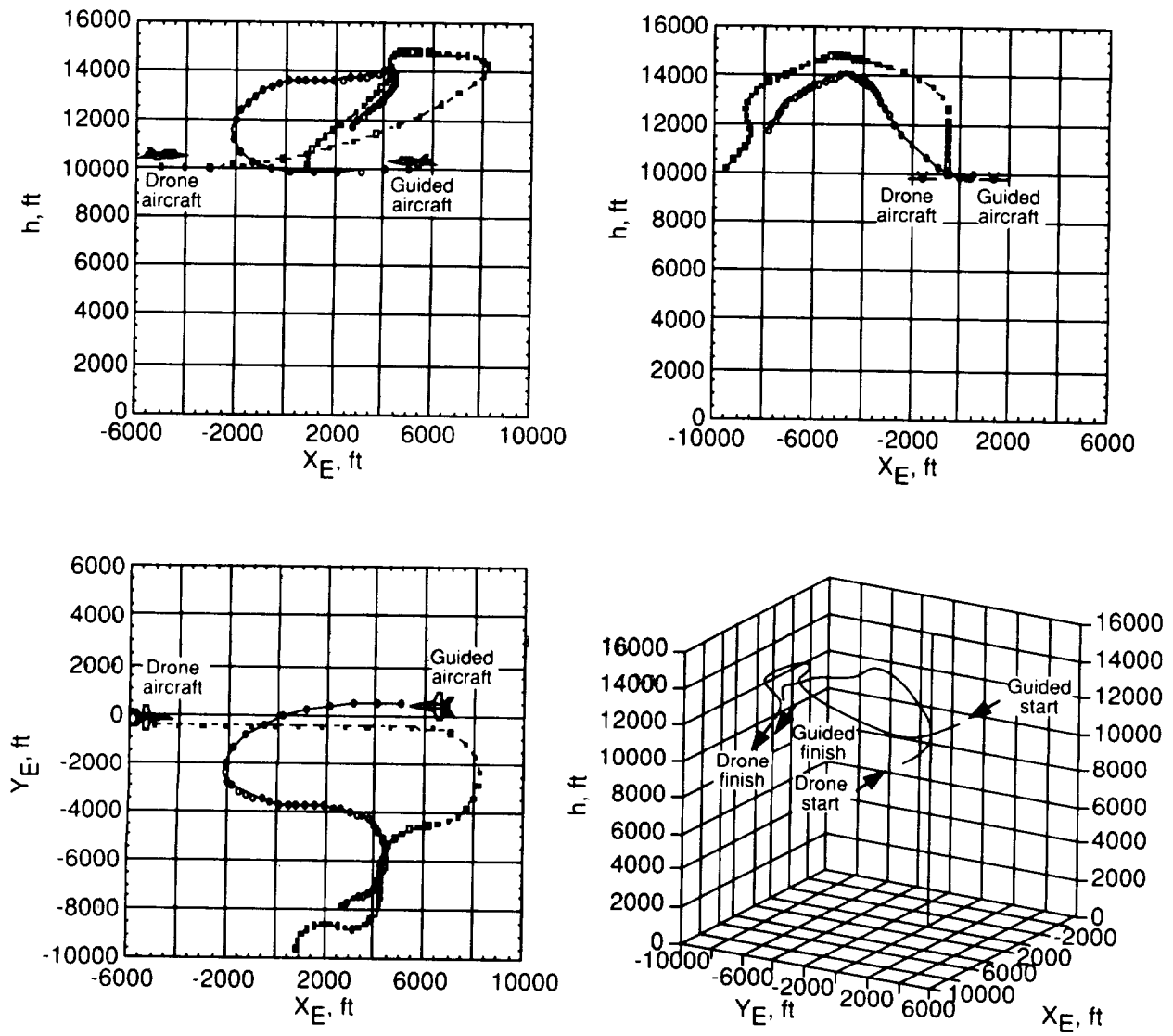
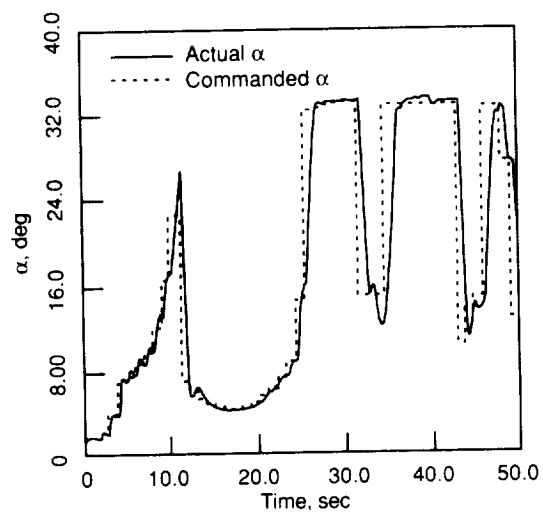
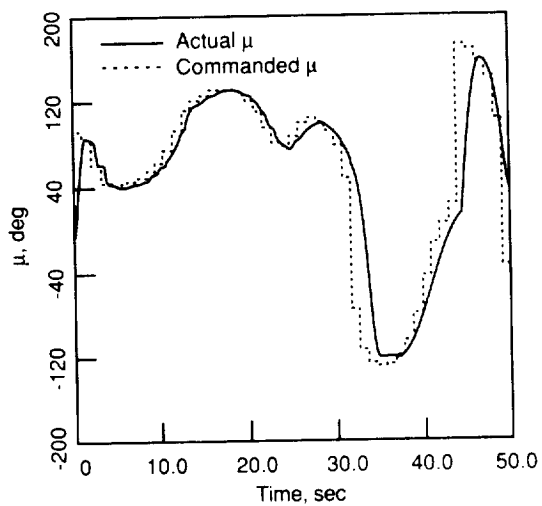


Figure 15. Example 1v1 engagement; positions plotted at 1-sec intervals.



(a) α response.



(b) Wind-axis bank-angle response.

Figure 16. Commanded and actual responses of guided aircraft.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY(Leave blank)	2. REPORT DATE July 1993	3. REPORT TYPE AND DATES COVERED Technical Memorandum		
4. TITLE AND SUBTITLE A High-Fidelity, Six-Degree-of-Freedom Batch Simulation Environment for Tactical Guidance Research and Evaluation		5. FUNDING NUMBERS WU 505-64-30-01		
6. AUTHOR(S) Kenneth H. Goodrich				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, VA 23681-0001		8. PERFORMING ORGANIZATION REPORT NUMBER L-17096		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001		10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA TM-4440		
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited Subject Category 08		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) A batch air combat simulation environment, the tactical maneuvering simulator (TMS), is presented. The TMS is a tool for developing and evaluating tactical maneuvering logics, but it can also be used to evaluate the tactical implications of perturbations to aircraft performance or supporting systems. The TMS can simulate air combat between any number of engagement participants, with practical limits imposed by computer memory and processing power. Aircraft are modeled using equations of motion, control laws, aerodynamics, and propulsive characteristics equivalent to those used in high-fidelity piloted simulations. Data bases representative of a modern high-performance aircraft with and without thrust-vectoring capability are included. To simplify the task of developing and implementing maneuvering logics in the TMS, an outer-loop control system, the tactical autopilot (TA), is implemented in the aircraft simulation model. The TA converts guidance commands by computerized maneuvering logics from desired angle of attack and wind-axis bank-angle inputs to the inner-loop control augmentation system of the aircraft. This report describes the capabilities and operation of the TMS and the TA.				
14. SUBJECT TERMS Air combat; Simulation; Agility; Tactics			15. NUMBER OF PAGES 51	
			16. PRICE CODE A03	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT	